

JUNIO DE 2014



GENERACIÓN AUTOMÁTICA DE NARRATIVAS

TRABAJO FIN DE GRADO

PABLO PÉREZ MORATO
UNIVERSIDAD CARLOS III DE MADRID

AGRADECIMIENTOS

En primer lugar, quiero mostrar mi agradecimiento a mis padres y mi hermano, que me han apoyado desde el principio en mi vida académica, primero disipando y resolviendo las dudas que pudiera tener y después apoyando cada decisión que he tomado en este sentido.

En segundo lugar, a mis amigos y compañeros de la universidad, por todos los momentos de camaradería y todas las ayudas que nos hemos prestado.

Al resto de mis amigos por compartir conmigo los momentos de tranquilidad y celebrar los momentos de éxito.

A Maite por presionarme constantemente para que trabajase en este proyecto todos los días y apoyarme en todo momento.

A mi tutor Daniel, por toda la ayuda prestada y por responder a todas mis dudas y preguntas a la mayor brevedad.

RESUMEN

En este trabajo se desarrolla un dominio de planificación de manera que, ejecutado en un planificador, se obtengan historias con interés narrativo. Para ello se hace uso del lenguaje PDDL y el planificador Sayphi. Este trabajo se centrará en primer lugar en las decisiones de diseño tomadas para modelar la información. En segundo lugar, se experimentará con el dominio desarrollado, utilizando para ello varios problemas de diferente complejidad y analizando los resultados obtenidos tanto desde el punto de vista de la planificación como desde el punto de vista narrativo.

ABSTRACT

In this bachelor thesis a planning domain is developed so that, executed in a planner, stories obtained could be interesting from a narrative point of view. To do that, the PDDL language and Sayphi planner are used. This paper will focus primarily on the design decisions taken to model the information. Second, the testing phase will take place, using problems with different complexity and analyzing the results from the point of view of planning and from the narrative point of view.

CONTENIDO

1.	INTRODUCCIÓN	1
2.	ESTADO DEL ARTE.....	3
2.1	PLANIFICACIÓN AUTOMÁTICA DE TAREAS	3
2.2	PDDL.....	4
2.3	SAYPHI.....	6
2.4	TRABAJOS SIMILARES	6
3.	OBJETIVO Y MOTIVACIÓN.....	8
4.	REQUISITOS	9
4.1	REQUISITOS FUNCIONALES	9
4.2	REQUISITOS NO FUNCIONALES	11
5.	DISEÑO DEL SISTEMA.....	12
5.1	DEFINICIÓN DE TIPOS DE OBJETOS	12
5.2	DEFINICIÓN DE PREDICADOS	14
5.3	DEFINICIÓN DE ACCIONES.....	16
5.3.1	<i>Acciones de asesinato.....</i>	<i>16</i>
5.3.2	<i>Acciones de movimiento y recogida de objetos.....</i>	<i>18</i>
5.3.3	<i>Acciones de robo y obtención de dinero</i>	<i>20</i>
5.3.4	<i>Acciones de la policía.....</i>	<i>21</i>
5.3.5	<i>Acciones emocionales.....</i>	<i>22</i>
5.3.6	<i>Acciones relacionadas con las organizaciones criminales</i>	<i>23</i>
5.4	DEFINICIÓN DE PROBLEMAS.....	23
5.5	CREACIÓN DE FRASES	27
6.	EXPERIMENTACIÓN	29
6.1	PROBLEMAS UTILIZADOS	29
6.1.1	<i>Elementos fijos: objetos.....</i>	<i>29</i>
6.1.2	<i>Elementos fijos: predicados</i>	<i>30</i>
6.1.3	<i>Elementos variables.....</i>	<i>31</i>
6.1.4	<i>Metas.....</i>	<i>32</i>
6.2	PLAN DE PRUEBAS	32
6.3	PROBLEMA 1.....	33
6.3.1	<i>Definición.....</i>	<i>33</i>
6.3.2	<i>Resultados</i>	<i>33</i>
6.4	PROBLEMA 2.....	34
6.4.1	<i>Definición.....</i>	<i>34</i>
6.4.2	<i>Resultados</i>	<i>34</i>
6.5	PROBLEMA3	36
6.5.1	<i>Definición.....</i>	<i>36</i>
6.5.2	<i>Resultados</i>	<i>36</i>
6.6	PROBLEMA 4.....	37
6.6.1	<i>Definición.....</i>	<i>37</i>
6.6.2	<i>Resultados</i>	<i>38</i>
6.7	PROBLEMA 5.....	39
6.7.1	<i>Definición.....</i>	<i>39</i>
6.7.2	<i>Resultados</i>	<i>39</i>
6.8	PROBLEMA 6.....	41
6.8.1	<i>Definición.....</i>	<i>41</i>

6.8.2	<i>Resultados</i>	42
6.9	PROBLEMA 7	44
6.9.1	<i>Definición</i>	44
6.9.2	<i>Resultados</i>	45
6.10	COMENTARIOS ACERCA DE LOS RESULTADOS	50
7.	GESTIÓN DEL PROYECTO	51
7.1	PLANIFICACIÓN INICIAL	51
7.2	PLANIFICACIÓN FINAL	52
7.3	PRESUPUESTO	53
8.	CONCLUSIONES	54
8.1	COMENTARIOS FINALES	54
8.2	LÍNEAS FUTURAS	54
9.	REFERENCIAS	56
ANEXOS I: MANUAL DE USUARIO		58
ANEXOS II: CÓDIGO		60
DOMINIO		60
PROBLEM1.PDDL		75
PROBLEM2.PDDL		77
PROBLEM3.PDDL		79
PROBLEM4.PDDL		81
PROBLEM5.PDDL		83
PROBLEM6.PDDL		85
PROBLEM7.PDDL		87

ÍNDICE DE TABLAS

TABLA 1. RF-001	9
TABLA 2. RF-002	9
TABLA 3. RF-003	9
TABLA 4. RF-004	9
TABLA 5. RF-005	9
TABLA 6. RF-006	9
TABLA 7. RF-007	10
TABLA 8. RF-008	10
TABLA 9. RF-009	10
TABLA 10. RF-010	10
TABLA 11. RF-011	10
TABLA 12. RF-012	10
TABLA 13. RNF-001.....	11
TABLA 14. RNF-002.....	11
TABLA 15. RNF-003.....	11
TABLA 16. RNF-004.....	11
TABLA 17. PREDICADOS.....	16
TABLA 18. ACCIONES DE ASESINATO	18
TABLA 19. ACCIONES DE MOVIMIENTO Y OBJETOS	19
TABLA 20. ACCIONES DE DINERO	21
TABLA 21. ACCIONES DE POLICÍA.....	22
TABLA 22. ACCIONES EMOCIONALES	22
TABLA 23. ACCIONES DE ORGANIZACIONES CRIMINALES.....	23
TABLA 24. FRASES DE ACCIONES.....	28
TABLA 25. PREDICADOS VARIABLES EN LOS PROBLEMAS.	32
TABLA 26. DATOS PROBLEMA 1	34
TABLA 27. DATOS PROBLEMA 2	35
TABLA 28. DATOS PROBLEMA 3	37
TABLA 29. DATOS PROBLEMA 4	39
TABLA 30. DATOS PROBLEMA 5	41
TABLA 31. DATOS PROBLEMA 6	44
TABLA 32. DATOS PROBLEMA 7	49
TABLA 33. GASTOS DE PERSONAL	53
TABLA 34. GASTOS DE MATERIAL AMORTIZABLE.	53
TABLA 35. PRESUPUESTO TOTAL.	53

ÍNDICE DE IMÁGENES

ILUSTRACIÓN 1. JERARQUÍA DE OBJETOS DEL DOMINIO.....	13
ILUSTRACIÓN 2. PLANIFICACIÓN INICIAL.	52
ILUSTRACIÓN 3. PLANIFICACIÓN FINAL.	52
ILUSTRACIÓN 4. RUTAS .SBCLRC	58
ILUSTRACIÓN 5. RUTAS .CLISPRC.....	58
ILUSTRACIÓN 6. RUTAS .INIT.....	58

1. INTRODUCCIÓN

Uno de los objetivos principales del uso de computadoras para la resolución de problemas es su capacidad para reducir significativamente el tiempo empleado a la vez que se evitan errores humanos. De esta forma, el tiempo ahorrado se puede invertir en otros procesos y se optimiza la producción. Uno de los tipos de procesos que resultan más beneficiados al ser resueltos por un computador es la planificación de tareas. La planificación consiste en, dado un estado inicial, ejecutar una serie de acciones de manera que en algún momento se satisfaga un estado final predefinido. Cuando los posibles estados y las acciones que se pueden ejecutar son numerosos, la planificación se vuelve imposible para un humano, por lo que se opta por el uso de máquinas, con una capacidad de cómputo mucho mayor.

La planificación automática de tareas es utilizada comúnmente para resolver problemas de la forma más rápida, sólida y eficiente posible, reduciendo el índice de variabilidad de las soluciones al mínimo. La particularidad de este Trabajo Fin de Grado es que se propone diseñar y modelar un dominio de planificación que, junto a un planificador, genere automáticamente textos narrativos variados e interesantes. Asimismo, a diferencia de otros trabajos de planificación, se pretende que la decisión de las acciones a realizar en cada instante tenga un índice de aleatoriedad, de manera que con el mismo estado inicial y estado final se generen diferentes planes.

La creación automática de textos narrativos puede tener gran importancia para autores y guionistas, al no haber intervención humana que pueda estar condicionada por trabajos anteriores propios o ajenos. Por ello, se podrán utilizar como base para construir historias aplicables a la novela, el cine, el teatro o los videojuegos. El nivel de detalle que es posible obtener con planificación automática puede llegar a ser muy elevado si se incluyen un gran número de acciones y estados, por lo que los planes pueden resultar muy interesantes. Otro ámbito de aplicación de la generación automática de historias es la creación de cuentos infantiles. Para ello sería posible definir varios géneros, de manera que se pudiera elegir qué tipo de cuento se desea obtener, los personajes que aparecerán y el final que tendrá. La diversidad entre los cuentos obtenidos es indispensable en este caso.

Para este Trabajo Fin de Grado se ha optado por generar textos de género negro en los que la figura de la mafia tenga una gran importancia, inspirados en el cine neo-noir surgido a partir de la década de los 70 (Chinatown, Los intocables de Elliot Ness, Muerte entre las flores, LA Condifential...) y en el cine de gánsters protagonizado por mafias (El Padrino, Uno de los nuestros, El precio del poder...). Este tipo de historias tienen un gran atractivo para el trabajo que nos ocupa por varias razones. En primer lugar, tienen un gran abanico de personajes entre los cuales hay una serie de relaciones muy variadas. Existen organizaciones criminales con jefes y miembros de las mismas, personas que pertenecen al departamento de policía, dueños de negocios y personas particulares. También son una parte importante de este tipo de tramas las relaciones afectivas, desencadenantes en ocasiones del resto de la acción (Muerte entre las flores). En segundo lugar, los personajes tienen una serie de objetivos que en muchas ocasiones estarán enfrentados. Los pertenecientes a una organización mafiosa actuarán al margen de la ley, lo cual intentarán evitar los agentes de la ley. Asimismo, los intereses entre las distintas

organizaciones mafiosas estarán enfrentados. Esto es especialmente interesante para la planificación de tareas, ya que una acción realizada por un personaje modifica el estado actual y afecta a otros personajes, de manera que se podrán ejecutar otras acciones de respuesta por parte de distintos personajes que a su vez afectarán notablemente al estado del problema. Por todo esto, el género comentado permite generar un dominio con un gran dinamismo.

En el este documento se presenta el desarrollo del dominio de planificación. En primer lugar, el apartado 2 habla de forma genérica acerca de la planificación, el lenguaje PDDL utilizado para modelar el dominio, y el planificador que se ha utilizado para obtener las historias, así como de trabajos similares que hayan sido hechos con anterioridad. A continuación se detallarán los objetivos del desarrollo de este dominio, así como los requisitos que tendrá que cumplir una vez terminado. En la sección 5 se ha documentado el diseño e implementación del dominio. Se explican con detalle las acciones y predicados más importantes, así como las decisiones de diseño tomadas. También se detallará el formato de los problemas que se utilizan. En la sección 6 se muestran los resultados de la fase de experimentación, en la que se definen varios problemas y, utilizando un planificador, se observa qué planes se obtienen, analizándose propiedades de estas como su longitud, diversidad de acciones y variabilidad entre distintos planes. A continuación se adjunta la planificación temporal del desarrollo del proyecto, así como la estimación de costes. Finalmente se exponen las conclusiones obtenidas tras el trabajo realizado y las posibles mejoras que se pueden implementar en un futuro.

2. ESTADO DEL ARTE

2.1 PLANIFICACIÓN AUTOMÁTICA DE TAREAS

La ingeniería informática, una disciplina relativamente nueva surgida de la necesidad de automatizar procesos anteriormente realizados por personas, ha experimentado un importante desarrollo durante las últimas décadas impulsada por la Tercera Revolución Industrial y produciendo lo que cada vez más comúnmente se llama “revolución informática”. Los grandes avances en este campo a medida que iba creciendo exponencialmente propiciaron que la disciplina se aplicara a situaciones cada vez más diversas. De esta forma, surgieron varias ramas, siendo la inteligencia artificial una de ellas.

La inteligencia artificial es la rama de la informática que permite crear sistemas que resuelvan problemas que resolvemos los humanos utilizando razonamiento. De esta manera, se han desarrollado sistemas expertos que sustituyen o sirven como apoyo a profesionales de ciertas áreas a la hora de resolver problemas cotidianos en sus puestos de trabajo; sistemas de análisis y síntesis del lenguaje; visión computerizada; sistemas de entretenimiento, etc. La rama de la inteligencia artificial que nos ocupa en el presente documento es la planificación automática de tareas.

Como ya se ha comentado anteriormente en este documento, la planificación automática de tareas es una disciplina de la inteligencia artificial que permite, dado un estado inicial y unas posibles acciones, llegar a un estado final definido con anterioridad [1]. El conjunto de acciones obtenido es el plan. En la planificación automática intervienen tres factores:

- **Dominio:** determina el escenario para el cual se van a obtener planes. Está formado principalmente por los predicados que pueden formar parte del estado en cada momento, y las posibles acciones que se pueden ejecutar.
- **Problema:** determina los objetos que aparecerán en el plan, el estado inicial formado por un conjunto de predicados, y el conjunto de metas que se pretende alcanzar con el plan.
- **Planificador:** es el software que genera un plan a partir de los datos del dominio y el problema.

La planificación automática es utilizada habitualmente en áreas tales como la robótica, la logística y la planificación temporal de acciones. Asimismo, la planificación puede utilizarse como una herramienta de análisis o puede aplicarse en tiempo real en un entorno profesional. Cabe destacar la existencia del Ames Planning and Scheduling Group de la NASA, un grupo de investigación y desarrollo de tecnología para la resolución de problemas complejos de planificación. En el caso de la NASA (aunque también en muchos otros tipos de aplicaciones prácticas), la mayoría de problemas que se deben resolver comparten una serie de particularidades que se controlan mediante planificación automática:

- Restricciones temporales complejas: algunos procesos deben realizarse en un orden concreto, y ciertas actividades sólo se pueden realizar en un periodo de tiempo muy concreto.
- Recursos limitados: las naves y vehículos utilizados en las misiones de la NASA tienen una limitación de recursos tales como la energía y la memoria. Por tanto, a la hora de realizar sus actividades, el gasto de estos recursos debe ser mínimo.
- Objetivos de diferente importancia: es común tener un gran número de objetivos a satisfacer, pero no todos tienen la misma importancia. La planificación otorgará prioridad a los de mayor importancia.
- Incertidumbre: no es posible determinar exactamente el tiempo requerido para desplazarse a otra posición.
- Autonomía: el plan obtenido debe poder ejecutarse de manera autónoma.

Algunas aplicaciones prácticas de las tecnologías desarrolladas por el Ames Planning and Scheduling Group son la creación de planificadores temporales para las tareas de la tripulación y del consumo de energía de las aeronaves, y el desarrollo de un sistema de planificación que sirva de ayuda al piloto en el caso de que la aeronave haya sufrido daños y deba aterrizar.

El evento más importante en el área que nos ocupa es la ICAPS (International Conference of Automated Planning and Scheduling), la conferencia anual de planificación automática. Este evento significa el mayor encuentro de investigadores de planificación automática, y cuenta con actividades relacionadas con este ámbito. Además, en el contexto de la ICAPS se organizan una serie de competiciones de planificación y aprendizaje automático tales como la International Planning Competition, orientada al desarrollo de sistemas de planificación de manera que sirvan para enfocar las investigaciones actuales en resolver y solucionar las limitaciones actuales.

2.2 PDDL

PDDL (Planning Domain Definition Language) es un lenguaje de programación, basado en STRIPS y ADL entre otros, desarrollado en 1998 en un intento de estandarizar los lenguajes de programación para planificación automática [2]. Su creación tuvo gran importancia a la hora de poder desarrollar la primera edición de la International Planner Competition (IPC), ya que facilitaba en gran medida la comparación entre los distintos sistemas de planificación. Por ello, ha sido el lenguaje oficial de todas las ediciones de la IPC.

Desde su creación, PDDL ha sido ampliado en varias ocasiones, añadiendo nuevas opciones:

- PDDL1.2: separó el dominio del problema en dos partes independientes.
- PDDL2.1: se incluyeron funciones numéricas, de manera que se pudieran modelar aspectos como el nivel de energía restante, distancia, tiempo... Asimismo, también se incluyeron *métricas* en los planes, de forma que éstas sirvan para evaluar la calidad de los planes. Las *métricas* pueden ser maximizadas o minimizadas para obtener los mejores planes posibles. También se añadieron *acciones durativas*. [3]

- PDDL2.2: en esta versión se incluyeron *predicados derivados* para modelar la dependencia entre dos hechos, y literales iniciales temporales, que permitirán la ejecución de ciertos eventos en un instante de tiempo concreto con independencia de la ejecución del plan. [4]
- PDDL3.0: se introdujeron *restricciones de transición de estados*, y *preferencias* que indican qué objetivos tienen mayor importancia, de manera que se pueda incluir en el problema una *métrica* que maximice la cantidad de objetivos preferentes que se alcanzan en un plan. [5]
- PDDL3.1: en la última versión de PDDL se incluyeron *funciones-objeto*, de manera que el valor de las funciones puede ser ahora de cualquier tipo-objeto que se haya definido.

Un dominio en PDDL1.2, que será la versión que se ha utilizado para implementar el trabajo que nos ocupa, tiene lasiguiente estructura:

En primer lugar se determina el nombre del dominio:

```
(define (domain depots)
```

A continuación se indican las características de PDDL utilizadas en el dominio, precedidas del carácter “:”.

```
(:requirements :strips :typing)
```

Se determina la jerarquía objetos que intervendrán en las acciones. *Object* es el objeto base del que heredan el resto.

```
(:types      place locatable - object
              depot distributor - place
              truck hoist surface - locatable
              pallet crate - surface)
```

Predicados que se utilizarán. La primera palabra representa el nombre del predicado, y los argumentos precedidos de “?” son variables del tipo cuyo nombre aparece a la derecha de cada uno (en el primer predicado del ejemplo, *?x* es una variable del tipo *locatable*).

```
(:predicates      (at ?x - locatable ?y - place)
                  (on ?x - crate ?y - surface))
```

Tras los predicados, se detallan las acciones del dominio. Una acción tiene una serie de parámetros, que serán las acciones que intervendrán en la misma

```
(:action Lift
  :parameters (?x - hoist ?y - crate ?z - surface ?p - place)
```

Para que la acción pueda ejecutarse, se deben cumplir las precondiciones:

```
:precondition      (and  (at ?x ?p)
                          (available ?x)
                          (at ?y ?p)
                          (on ?y ?z)
                          (clear ?y))
```

Finalmente, se indica el efecto de la acción, que supone una modificación en el estado actual:

```
:effect      (and  (not (at ?y ?p))
                   (lifting ?x ?y)
                   (not (clear ?y))
                   (not (available ?x))
                   (clear ?z)
                   (not (on ?y ?z))))
```

2.3 SAYPHI

Sayphi es un sistema de planificación que, tomando como base un planificador basado en heurísticas como FF [6], incluye técnicas de aprendizaje automático para mejorar los resultados [7]. Sayphi es un planificador independiente del dominio, de forma que su código no hace referencia a ningún dominio en particular, por lo que puede generar planes en cualquier dominio. En este proyecto se ha trabajado con una versión de este planificador que permite la introducción de cierta aleatoriedad en las decisiones. Así, permite planificación:

- Aleatoria: la acción que se escoge en cada momento se elige de manera aleatoria entre las que se pueden ejecutar por sus precondiciones. Debido a la mayor dificultad de obtener planes con una ejecución aleatoria, es posible determinar el número de iteraciones de este proceso, obteniendo un plan válido si es posible.
- No aleatoria: la acción escogida a cada momento vendrá determinada por los mecanismos de planificación implementados. Es posible determinar la probabilidad de que la acción se escoja de manera aleatoria utilizando el parámetro *probability-ehc* (valores de 0 a 1). De esta forma, escogiendo el valor 0 el plan será totalmente aleatorio, y escogiendo el valor 1 no habrá ninguna aleatoriedad y el plan obtenido será el mismo en cada ejecución.

2.4 TRABAJOS SIMILARES

Pese a que la generación de historias es una aplicación de la planificación menos común que otras como la robótica, en la última década se han publicado numerosos estudios acerca de maneras de mejorar las historias obtenidas mediante esta técnica. Estas investigaciones han tenido dos principales objetivos: obtener el mejor modelado de la información del dominio, y mejorar los mecanismos de control de la narración en el planificador.

El trabajo más parecido a lo que se ha realizado en este proyecto es el descrito en “Narrative Planning: Compilations to Classical Planning” [8]. En este trabajo se busca generar narrativas mediante planificación, diferenciando entre las metas que se desea obtener de las intenciones de los personajes, las cuales podrán cambiar durante el transcurso de la historia. Asimismo, en este trabajo P. Haslum añade a las acciones que ejecutan los personajes una intención previa, de manera que estas acciones solo se puedan ejecutar cuando hay una intención.

En el artículo “Planning characters’ behaviour in interactive storytelling” [9], se comenta su implementación de HTN (redes jerárquicas de tareas) en la creación de historias, de manera que a cada personaje que interviene en la trama le corresponde un HTN. La tarea principal que deberá realizar cada personaje se descompondrá en varias subtarefas, de manera que en los niveles más bajos cada nodo representará una forma de alcanzar los objetivos del personaje. También se implementan mecanismos de replanificación en caso de que acciones que se encontraban en la HTN de un personaje no puedan ejecutarse debido al desarrollo del plan.

Por el contrario, en el artículo “Applying Planning to Interactive Storytelling: Narrative Control using State Constraints” [10] se propone enfocar las investigaciones en obtener un modelado del dominio más completo. En este estudio se opta por utilizar las *restricciones de transición de estados* (restricciones fuertes) y las *preferencias* (restricciones débiles) de PDDL3.0, de manera que en los planes obtenidos se deban respetar todas las restricciones fuertes y se deba satisfacer el mayor número de restricciones débiles.

Cabe destacar que, a diferencia del presente trabajo, muchos de los estudios acerca de *interactive storytelling* tienen como objetivo obtener planes en los que el usuario tenga cierta capacidad de decisión e influya en la historia.

3.OBJETIVO Y MOTIVACIÓN

Como ya se ha comentado anteriormente, el objetivo de este trabajo es la creación de un dominio que, ejecutado en un planificador junto con una serie de problemas, genere narrativas. Se perseguirá que las narrativas obtenidas sean lo más extensas posibles e incluyan el mayor número de acciones distintas. Además, deben tener cierta lógica y ser interesantes para el usuario que vaya a utilizar el sistema. Se buscará que los planes obtenidos tengan cierta aleatoriedad, de manera que varias ejecuciones con la misma configuración generen distintas historias.

Para obtener planes interesantes y con cierto grado de complejidad, el número de acciones y la cantidad de elementos que influyen en las mismas serán muy altos, por lo que el tiempo empleado para resolver los problemas puede llegar a ser elevado. Debido a esto, uno de los objetivos del trabajo será optimizar el modelado del dominio de manera que emplee el mínimo tiempo en resolver los problemas definidos. Para probar que el modelado del dominio es lo suficientemente complejo, se utilizarán problemas en los que intervengan un número considerable de personajes (entre 6 y 10), y entre los cuales existan relaciones de distinto tipo. Asimismo, en cada problema habrá varias metas, las cuales no serán triviales de obtener.

Se buscará también maximizar la escalabilidad del dominio, de manera que pueda ser ejecutado en el mayor número de planificadores posible y en todas las plataformas. Para ello, se modelará el conocimiento utilizando PDDL1.2, cuyas funcionalidades son soportadas por la gran mayoría de planificadores.

La motivación principal es que los planes obtenidos puedan ser utilizados como inspiración para la construcción de historias aplicables en la industria del cine, la televisión, el teatro, la narrativa o los videojuegos. Asimismo, se propone una forma de mejorar la obtención de historias mediante planificación de tareas sin necesidad de modificar o crear un planificador, sino mejorando el modelado del dominio.

4. REQUISITOS

El sistema completo, incluyendo el planificador, el dominio y los problemas creados, tiene una serie de requisitos funcionales y no funcionales que deben cumplirse.

4.1 REQUISITOS FUNCIONALES

RF-001: Ejecución con dominio y problema			
Descripción	El planificador podrá ejecutar utilizando un dominio y un problema escritos en PDDL.		
Prioridad	Alta	Necesidad	Alta

Tabla 1. RF-001

RF-002: Ejecución aleatoria			
Descripción	El sistema se podrá ejecutar en modo aleatorio.		
Prioridad	Alta	Necesidad	Alta

Tabla 2. RF-002

RF-003: Modificar número de iteraciones			
Descripción	El usuario podrá modificar el número de iteraciones de ejecución en modo aleatorio		
Prioridad	Baja	Necesidad	Baja

Tabla 3. RF-003

RF-004: Ejecución mediante heurísticas			
Descripción	El sistema se podrá ejecutar en modo de planificación mediante heurísticas		
Prioridad	Alta	Necesidad	Alta

Tabla 4. RF-004

RF-005: Dominio y problema escritos en PDDL1.2.			
Descripción	El dominio y el problema estarán escritos en PDDL1.2.		
Prioridad	Alta	Necesidad	Alta

Tabla 5. RF-005

RF-006: Definición de nuevos problemas			
Descripción	El usuario podrá definir nuevos problemas		
Prioridad	Alta	Necesidad	Alta

Tabla 6. RF-006

RF-007: Elección de problema para ejecutar			
Descripción	El usuario podrá determinar qué problema se desea resolver en la ejecución		
Prioridad	Alta	Necesidad	Alta

Tabla 7. RF-007

RF-008: Elección de tiempo límite			
Descripción	El usuario podrá determinar el tiempo límite para obtener un plan.		
Prioridad	Baja	Necesidad	Media

Tabla 8. RF-008

RF-009: Solución			
Descripción	El sistema mostrará la solución en la consola de comandos.		
Prioridad	Alta	Necesidad	Alta

Tabla 9. RF-009

RF-010: Avisar plan fallido			
Descripción	El sistema indicará en la consola de comandos cuando no sea posible obtener un plan.		
Prioridad	Alta	Necesidad	Alta

Tabla 10. RF-010

RF-011: Datos de creación de plan			
Descripción	El sistema mostrará tras la ejecución el tiempo empleado, los nodos visitados y la longitud del plan.		
Prioridad	Baja	Necesidad	Media

Tabla 11. RF-011

RF-012: Modificar frases de salida			
Descripción	El usuario podrá modificar las frases de salida que representan las acciones del dominio.		
Prioridad	Baja	Necesidad	Baja

Tabla 12. RF-012

4.2 REQUISITOS NO FUNCIONALES

RNF-001: Plataformas			
Descripción	El sistema se podrá ejecutar en sistemas operativos Windows, Linux y Mac OS X.		
Prioridad	Alta	Necesidad	Alta

Tabla 13. RNF-001

RNF-002: Tiempo límite			
Descripción	El sistema no tardará más del tiempo límite definido por el usuario en obtener un plan.		
Prioridad	Alta	Necesidad	Alta

Tabla 14. RNF-002

RNF-003: Limitaciones de PDDL1.2			
Descripción	El sistema no permitirá la inclusión de fluents ni acciones durativas.		
Prioridad	Alta	Necesidad	Alta

Tabla 15. RNF-003

RNF-004: Iniciar SBCL			
Descripción	El usuario debe haber iniciado sesión en SBCL antes de ejecutar el sistema.		
Prioridad	Alta	Necesidad	Alta

Tabla 16. RNF-004

5. DISEÑO DEL SISTEMA

Como se ha comentado anteriormente, el objetivo de este trabajo es obtener historias de longitud considerable variadas e interesantes. Para ello, se ha dedicado especial esfuerzo en obtener el mejor modelado del dominio posible. Se ha decidido que las historias generadas tengan como escenario el género negro o policíaco, de manera que existan elementos como la mafia, la policía, los asesinatos, los secuestros, los robos, la extorsión y las relaciones sentimentales. Este género narrativo permite una gran versatilidad a la hora de crear historias y de modelar el dominio, ya que existen multitud de personajes con diferentes motivaciones, las cuales estarán en muchos casos enfrentadas.

Esta sección se dividirá en cinco apartados diferentes. El primero estará dedicado a la presentación de los objetos que intervendrán en la historia. En el segundo de los apartados se comentarán los predicados incluidos en el dominio y se defenderán las decisiones de diseño tomadas referentes a los mismos. Después se describirán las acciones que pertenecen al dominio. A continuación se definirá el formato de los problemas y las condiciones que deben cumplir. Finalmente se explicarán las modificaciones llevadas a cabo en el código del planificador para la obtención de oraciones que representen cada acción.

5.1 DEFINICIÓN DE TIPOS DE OBJETOS

Es de gran importancia para optimizar el modelado del problema lograr una estructura de objetos altamente jerarquizada, así como utilizar correctamente dichos objetos en las acciones y en los predicados. El uso del nivel de jerarquía más profundo posible en predicados y acciones reduce el número de instancias innecesarias de una acción que se pueden ejecutar, haciendo más rápida la planificación.

El objeto principal, y que intervendrá en la mayoría de acciones, ya que es el motor de los acontecimientos que ocurrirán en la historia, es *person*. A su vez, de *person* heredan *male* y *female*, ya que en determinadas acciones conviene diferenciar entre ambos géneros. *Person*, junto con *weapon*, hereda de *locatable*. *Locatable* será el único objeto físico que está asociado constantemente con un lugar, de manera que pueda cambiar de sitio.

Los lugares por los que se moverán los personajes y en los cuales transcurrirá la acción también se dividen en varios subtipos. El tipo básico es *place*, en el que se incluyen *jail*, *genericplace* y *kidnapplace*. Salvo *genericplace*, los otros dos subtipos no pueden ser visitados por ninguna persona de manera normal, sino que llegarán a ellos como consecuencia de acciones concretas. *Jail* representa la cárcel, a la que los personajes pueden llegar si les arresta la policía, y de la cual únicamente se puede salir escapando ilegalmente. Cabe destacar que se ha decidido modelar la cárcel como un lugar, en vez de utilizar un predicado que represente que un personaje está encarcelado, para poder reutilizar el predicado “*at*”. *Kidnapplace* representa el lugar en el que se encuentra una persona mientras está secuestrada, y la única manera de salir es pagando el rescate.

Por el contrario, *genericplace* representa el tipo de lugares que los personajes pueden visitar por su propia voluntad y donde transcurre la acción. A su vez, se divide en cuatro

subtipos: *business* representa un negocio de un particular, como puede ser un restaurante o una joyería; *bank* es un banco que puede ser robado, *otherplace* abarca todos los lugares que no tienen otra particularidad, como una casa, un parque, una estación de trenes, etc, y *policestation* representa la central de policía a la que se llegará como resultado de acciones concretas. Se han modelado también como objetos los motivos por los cuales se puede producir un asesinato: *vengeance*, *jealousy* y *treason*. Estos tres objetos heredan de *motive*.

Cabe destacar que otra importante decisión de diseño ha sido incluir los distintos niveles de dinero que puede tener cada personaje como *tipos*, en vez de utilizar *fluents*. De esta forma, tenemos cuatro niveles distintos, en orden ascendente: *nomoney*, *lowmoney*, *mediummoney* y *highmoney*. Todos son subtipos de *moneylevel*.

En el siguiente gráfico se muestra la jerarquía de objetos del dominio:

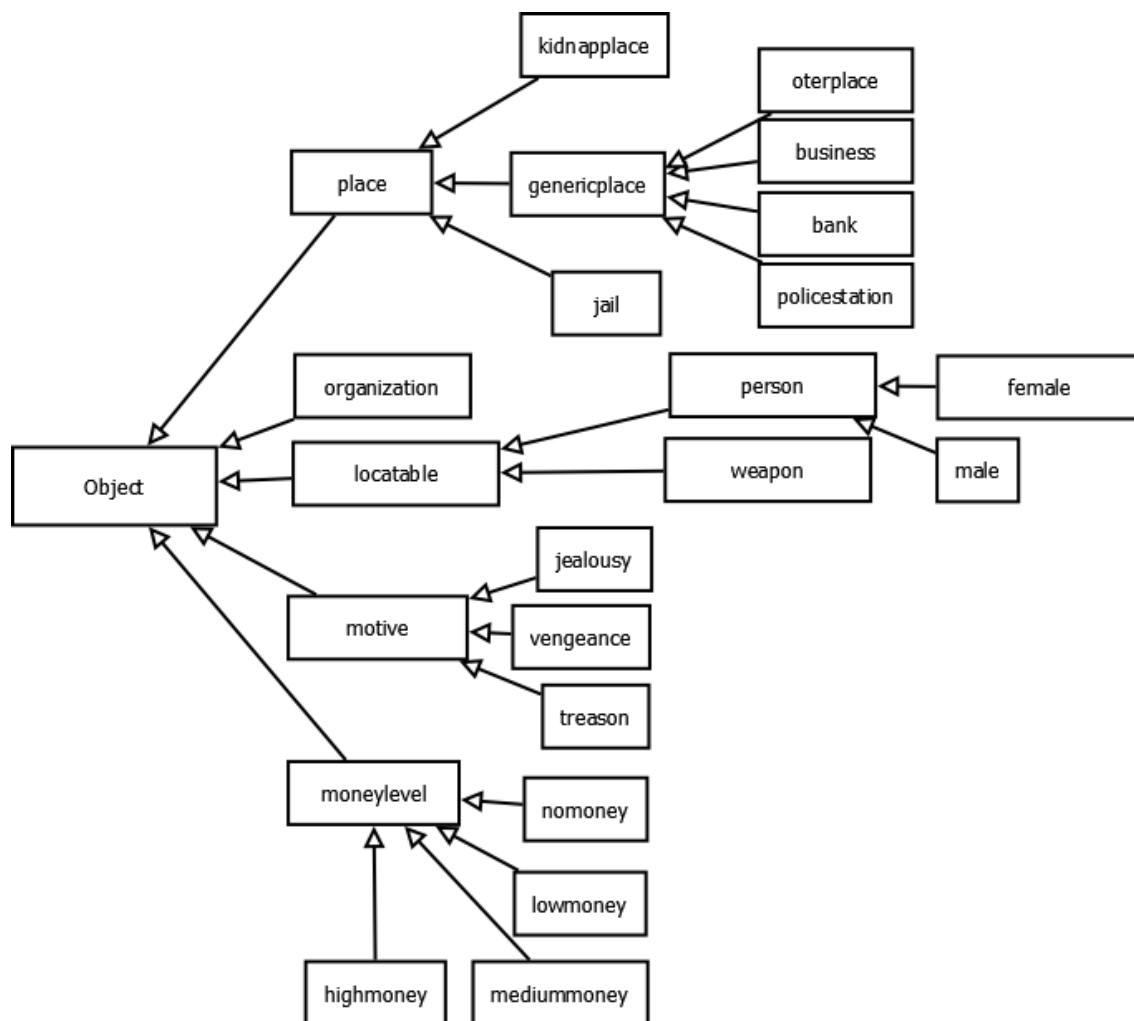


Ilustración 1. Jerarquía de objetos del dominio

5.2 DEFINICIÓN DE PREDICADOS

Se ha modelado como predicados del dominio información de distintos tipos. Los predicados más básicos son referentes al estado de los personajes, de manera que se utilizan para representar en qué lugar se encuentra cada uno, si está vivo o muerto, si está herido o si por alguna circunstancia no puede desplazarse (predicado *non-movable*). Esta circunstancia no era necesario modelarla, ya que se podría haber representado con los predicados que indican que esté muerto, secuestrado o encarcelado, pero la posibilidad de que un personaje esté inmóvil por tres motivos diferentes añadía complejidad al dominio. De esta forma, en las precondiciones de la mayoría de acciones se incluye que el personaje que las realiza no puede estar inmóvil, sin necesidad de incluir que debe estar vivo, no secuestrado y no encarcelado.

También se modelan como predicados los estados relacionados con los objetos. Una persona puede tener un objeto / arma. De esta forma se activarían dos predicados: (*has-weapon* ?p1 - person) y (*has-object* ?p1 - person ?weapon - weapon). La razón de utilizar dos predicados distintos es la de optimizar la cantidad de nodos que explora el planificador al ejecutar cada acción. Hay determinadas acciones que requieren que el personaje esté armado, pero no es necesario saber qué arma es. Si se utilizara el predicado *has-object*, habría que definir el arma como parámetro de la acción, y el número de posibilidades de ejecutar dicha acción se multiplicaría por el número de armas que hubiera instanciadas en el problema. Por ello, en este tipo de predicados se utiliza *has-weapon*. Además, también es posible que haya una prueba delictiva en un arma.

Se han incluido predicados referentes también a las organizaciones criminales. Estas organizaciones de tipo mafioso tienen un líder y pueden tener también un sucesor del líder, que asumirá el puesto en caso de fallecimiento del primero. También hay una serie de personajes que pertenecen a una organización. Además, se ha modelado la posibilidad de que un personaje sea el guardaespaldas de otro. Como ocurría con las armas, se han creado los predicados genéricos *is-boss* y *has-bodyguard*, además de aquellos en los que se determina de qué organización es líder y qué personaje es guardaespaldas de otro.

También se han modelado las relaciones personales mediante predicados. De esta forma podrá haber amantes, a un personaje le podrá gustar otro, y se ha añadido la posibilidad de que un personaje haya besado a otro. Asimismo, habrá padres e hijos.

Existen también una serie de predicados dedicados a los actos delictivos. Se ha modelado el hecho de que un personaje tenga un motivo para asesinar a otro, que tenga la intención de asesinarlo o que se le haya ordenado asesinarlo. Asimismo, se puede tener la intención de secuestrar a otro personaje. Los personajes que sean policías (definido mediante el predicado *is-police*) podrán además tener la intención de capturar a otro si ha cometido crimen. Se han utilizado predicados también para modelar hechos que ocurrieron en el pasado y que se utilizarán como precondiciones en algunas acciones: *robbed* y *killed*.

Uno de los elementos más importantes que intervendrán en la historia es el dinero y los negocios. Se ha modelado la cantidad de dinero que tiene un personaje con el predicado (*has-money-level* ?person - person ?level - moneylevel), y en cada problema se incluye la jerarquía entre los distintos niveles de dinero utilizando el predicado (*next-money-level* ?l1 - moneylevel ?l2 - moneylevel). Se

ha añadido la posibilidad de que la motivación de un personaje en la historia sea obtener dinero, y de que haya una deuda monetaria entre dos personajes. Asimismo, un personaje podrá ser dueño de un negocio, el cual podrá pasar a estar destruido a lo largo de la historia.

En la siguiente tabla se muestran todos los predicados incluidos.

Predicado	Parámetros	Descripción
At	?l - locatable ?place - place	El personaje o objeto se encuentra en un lugar.
Injured	?p1 - person	El personaje está herido.
Dead	?p - person	El personaje está muerto.
Non-movable	?p - person	El personaje está inmóvil (generalización para muerto, secuestrado o encarcelado).
Likes	?p1 - person ?p2 - person	A ?p1 le gusta ?p2.
Lovers	?p1 - person ?p2 - person	Los dos personajes son amantes.
Kissed	?p1 - person ?p2 - person	?p1 besó a ?p2.
Father-son	?p1 - person ?p2 - person	?p1 es el padre de ?p2.
Has-weapon	?p1 - person	El personaje tiene un arma, sin especificar cuál.
Has-object	?p1 - person ?weapon - weapon	El personaje tiene un arma concreta.
Clue-in-object	?person - person ?weapon - weapon	Hay una prueba delictiva de ?person en el arma.
Is-police	?p1 - person	El personaje es policía.
Suspecting-about	?p1 - person ?p2 - person	?p1 sospecha que ?p2 ha cometido un delito.
Has-motive	?p1 - person ?p2 - person ?motive - motive	?p1 tiene un motivo para vengarse de ?p2.
Commanded-to-kill	?killer - person ?victim - person ?commandedkiller - person	?commandedkiller ha ordenado a ?killer matar a ?victim.
Intends-to-kill	?killer - person ?victim - person	?killer pretende matar a ?victim.
Intends-to-capture	?p1 - person ?p2 - person	?p1 pretende capturar a ?p2 y encarcelarlo.
Intends-to-kidnap	?kidnapper - person ?victim - person	?kidnapper pretende secuestrar a ?victim.
Kidnapped-by	?p1 - person ?p2 - person	?p1 está secuestrado por ?p2.
Killed	?killer - person ?victim - person	?killer ha matado a ?victim.
Robbed	?p1 - person ?p2 - person	?1 ha robado en el negocio de ?p2.
Boss-organization	?p1 - person ?or - organization	El personaje es jefe de la organización ?or.

Is-boss	?p1 - person	El personaje es líder de una organización no concretada.
Successor-organization	?p1 - person ?or - organization	El personaje es el sucesor al poder de la organización ?or.
In-organization	?p1 - person ?or - organization	El personaje es miembro de la organización ?or.
Bodyguard	?p1 - person ?p2 - person	?p1 es el guardaespaldas de ?p2.
Has-bodyguard	?p1 - person	El personaje tiene un guardaespaldas.
Has-business	?p1 - person ?b - business	El personaje tiene el negocio ?b.
destroyed	?b - business	El negocio ha sido destruido.
Debt	?p1 - person ?p2 - person	?p1 tiene una deuda con ?p2.
Has-money-level	?person - person ?level - moneylevel	El personaje tiene un cierto nivel de dinero.
Next-money-level	?l1 - moneylevel ?l2 - moneylevel	?l2 es el siguiente nivel de dinero a ?l1.
Motivation-money	?person - person	El personaje tiene la motivación de obtener dinero.

Tabla 17. Predicados.

5.3 DEFINICIÓN DE ACCIONES

El conjunto de acciones constituye el apartado más importante del dominio. Una definición de acciones variada, con precondiciones relevantes y que afecte a un número considerable de predicados del escenario es indispensable para la construcción de historias interesantes. Es posible dividir las acciones del dominio que nos ocupa en varios bloques.

5.3.1 Acciones de asesinato

El asesinato, junto al robo y la obtención de dinero, es uno de los motores de las historias que se generarán con este dominio.

En primer lugar, que un personaje tenga un motivo para asesinar a otro desembocará en que decida matarlo por su cuenta si el primero no es un líder de una organización criminal, o por el contrario que ordene el asesinato a un miembro de la organización. Esto se determina en las acciones *decide-murder* y *order-murder*. Una vez se ha desarrollado la intención o la orden, las acciones de asesinato pueden ser ejecutadas. Existen cuatro acciones principales, dependiendo de si el asesinato es una orden o no y de si la potencial víctima tiene un guardaespaldas: *kill*, *kill-by-order*, *kill-bodyguard* y *kill-bodyguard-by-order*. Cabe destacar que se ha modelado que una víctima no puede ser atacada mientras tenga un guardaespaldas, por lo que si se pretende matar a un personaje con guardaespaldas, se ejecutará antes la acción de matar a este último. También se han incluido *failed-kill* y *failed-kill-by-order*, que añaden la posibilidad de que un intento de asesinato falle, dejando a la víctima potencial herida y adquiriendo un motivo de venganza sobre el responsable del ataque.

A continuación se muestra y se explica la acción *kill-by-order*, representativa de este tipo de acciones.

```
(:action kill-by-order
  :parameters (?killer ?victim ?boss - person ?place - genericplace
    ?weapon - weapon)
  :precondition (and
    (not(non-movable ?killer))
    (not(non-movable ?victim))
    (not(non-movable ?boss))
    (ordered-to-kill ?killer ?victim ?boss)
    (not(= ?killer ?boss))
    (at ?killer ?place)
    (at ?victim ?place)
    (has-object ?killer ?weapon)
    (not(has-bodyguard ?victim)))
  :effect (and
    (dead ?victim)
    (non-movable ?victim)
    (clue-in-object ?killer ?weapon)
    (not(ordered-to-kill ?killer ?victim
      ?boss))
    (killed ?boss ?victim)))
```

Para que se ejecute, el jefe de organización criminal debe haber ordenado a un miembro que asesine a otra persona. Asimismo, ninguno de los tres implicados puede estar inmóvil. Además, la víctima no puede tener guardaespaldas, y debe estar en el mismo lugar que el asesino, el cual debe llevar un arma. Los efectos de la acción son varios. En primer lugar, la víctima pasará a estar muerta e inmóvil, por lo que no podrá realizar más acciones. Además, en el arma utilizada pasará a haber una prueba delictiva que la policía podrá utilizar para encarcelar al asesino (y no al responsable final del asesinato: el que lo ordenó). Finalmente, se eliminará el predicado *ordered-to-kill* y se añadirá (*killed ?boss ?victim*), mediante el cual la policía sí podrá sospechar del jefe de la organización criminal como responsable. El resto de acciones de asesinato serán similares, con pequeñas diferencias ya comentadas anteriormente.

Acción	Descripción
Decide-murder	Un personaje con un motivo de asesinato sobre otro decide asesinarlo.
Command-murder	Un líder de organización con un motivo de asesinato sobre otro ordena a un miembro de su organización asesinarlo.
Kill	Un personaje mata a otro por su propia cuenta.
Kill-by-command	Un personaje mata a otro por orden del líder de su organización.
Kill-bodyguard	Un personaje mata al guardaespaldas de la persona a la que quiere asesinar por su propia cuenta.
Kill-bodyguard-by-command	Un personaje mata al guardaespaldas de la persona a la que quiere asesinar por su orden del líder de su organización.
Failed-kill	Un personaje falla al intentar asesinar a otro por su propia cuenta, quedando este último herido.
Failed-kill-by-command	Un personaje falla al intentar asesinar a otro por orden del líder de su organización, quedando este último herido.

Vengeance-lover	Un personaje adquiere un motivo de venganza sobre otro por el asesinato de su amante.
Vengeance-father	Un personaje adquiere un motivo de venganza sobre otro por el asesinato de su padre.

Tabla 18. Acciones de asesinato

5.3.2 Acciones de movimiento y recogida de objetos

En las historias que se generen mediante el sistema de planificación el movimiento de los personajes por los distintos lugares estará siempre motivado por sus intenciones. Para modelar esto se han generado 11 acciones distintas referentes al movimiento:

- Move-to-kill
- Move-to-kill-byorder
- Move-to-kill-bodyguard
- Move-to-kill-bodyguard-by-order
- Move-to-pickup-object-kill
- Move-to-pickup-object-kill-by-order
- Move-to-kidnap
- Move-to-rob-bank
- Move-to-rob-business
- Move-to-pick-up-object-rob
- Move-to-kiss

La gran mayoría de las acciones de este tipo tienen la siguiente estructura. Un personaje con una motivación está en un lugar concreto. El objeto de dicha motivación, ya sea una persona o un arma, se encuentra en otro lugar distinto. Como efecto, la persona se traslada al segundo lugar, y en el caso de las acciones del tipo *move-to-pick-up-** recoge también el objeto. Además, se han incluido también acciones del tipo *pick-up-object*, en las que para recoger un objeto no es necesario desplazarse. Esto ayudará a mejorar el realismo de las narrativas obtenidas, ya que un personaje recogerá antes un arma que se encuentre en el mismo lugar que él que una que no lo está. A continuación se muestra *move-to-rob-business* como ejemplo de este tipo de acciones.

```
(:action move-to-rob-business
  :parameters (?robber ?victim - person
               ?place1 - genericplace ?business - business)
  :precondition (and
    (at ?robber ?place1)
    (not(non-movable ?robber))
    (not(dead ?victim))
    (has-weapon ?robber)
    (not(destroyed ?business))
    (has-business ?victim ?business)
    (motivation-money ?robber))
  :effect (and
    (not(at ?robber ?place1))
    (at ?robber ?business)))
```

De esta acción es preciso comentar que un personaje que no tiene la motivación de obtener dinero no se desplazará a un negocio para robarlo, evitando viajes innecesarios. Asimismo, para realizar el viaje debe tener ya un arma.

Acción	Descripción
Move-to-kill	Un personaje se mueve al lugar donde está otro al que tiene intención de matar.
Move-to-kill-by-command	Un personaje se mueve al lugar donde está otro al que se le ha ordenado matar.
Move-to-kill-bodyguard	Un personaje se mueve al lugar donde está el guardaespaldas del personaje al que tiene intención de matar.
Move-to-kill-bodyguard-by-command	Un personaje se mueve al lugar donde está el guardaespaldas del personaje al que se le ha ordenado matar.
Move-to-kidnap	Un personaje se mueve al lugar donde está otro al que intención de secuestrar.
Move-to-rob-bank	Un personaje se mueve a un banco para robarlo.
Move-to-rob-business	Un personaje se mueve a un negocio para robar en él.
Move-to-kiss	Un personaje masculino se mueve al lugar en el que está otro femenino que le gusta.
Pick-up-object-to-kill	Un personaje desarmado con intención de matar a otro coge un objeto que se encuentra en su misma posición.
Pick-up-object-to-kill-by-command	Un personaje desarmado con orden de matar a otro coge un objeto que se encuentra en su misma posición.
Pick-up-object-to-rob	Un personaje desarmado con intención de obtener dinero coge un objeto que se encuentra en su misma posición.
Move-to-pickup-object-kill	Un personaje desarmado con intención de matar a otro se mueve a un lugar donde hay un objeto y lo coge.
Move-to-pickup-object-kill-by-command	Un personaje desarmado con orden de matar a otro se mueve a un lugar donde hay un objeto y lo coge.
Move-to-pickup-object-rob	Un personaje desarmado con intención de obtener dinero se mueve a un lugar donde hay un objeto y lo coge.
Drop-object	Un personaje que tiene un objeto lo suelta.

Tabla 19. Acciones de movimiento y objetos

5.3.3 Acciones de robo y obtención de dinero

El dinero es otro factor que hará que las historias sean interesantes. Existen multitud de acciones relacionadas con el dinero en el dominio. En primer lugar se incluye la posibilidad de haber deudas entre personajes. Una deuda se producirá cuando un personaje le haya hecho un préstamo a otro. Las consecuencias son varias. Los personajes pueden pagar deudas si tienen el nivel de dinero necesario, y en caso contrario pueden ser secuestrados. El secuestro tiene como fin que un allegado a la víctima pague el rescate, con lo que la deuda estaría saldada. Por otro lado, existen personajes cuya motivación principal es obtener dinero, modelado como un predicado. Estos personajes podrán robar negocios y bancos hasta que alcancen el máximo nivel de dinero, tras lo cual perderán dicha motivación.

El paso de dinero entre los personajes se ha modelado utilizando los predicados *has-money-level* y *next-money-level*. De esta forma, en una acción en la que un personaje roba en el negocio de otro, como precondition se establece que el ladrón tiene un nivel de dinero que no es el máximo, y que la víctima tiene un nivel de dinero que no es el mínimo. De esta forma, en los efectos de la acción se incluye que el ladrón pasa a tener el nivel siguiente de dinero al que tenía, y la víctima pasa a tener el nivel anterior. A continuación se muestra la acción *rob-business*, ejemplo de las acciones en las que tiene influencia el dinero.

```
(:action rob-business
  :parameters (?robber ?victim - person ?business - business
    ?vengeance - vengeance ?robberl1 ?robberl2 ?victiml1
    ?victiml2 - moneylevel)
  :precondition (and
    (at ?robber ?business)
    (not(non-movable ?robber))
    (not(dead ?victim))
    (has-business ?victim ?business)
    (next-money-level ?robberl1 ?robberl2)
    (next-money-level ?victiml2 ?victiml1)
    (has-money-level ?robber ?robberl1)
    (has-money-level ?victim ?victiml1)
    (not(robbed ?robber ?victim)))
  :effect (and
    (has-motive ?victim ?robber ?vengeance)
    (not(has-money-level ?robber ?robberl1))
    (not(has-money-level ?victim ?victiml1))
    (has-money-level ?robber ?robberl2)
    (has-money-level ?victim ?victiml2)
    (robbed ?robber ?victim)))
```

Acción	Descripción
Loan	Un personaje con el máximo nivel de dinero le presta un nivel a otro que tiene el mínimo.
Pay-debt	Un personaje endeudado con otro paga al segundo su deuda.
Decide-kidnap-father-son	Un personaje al cual se le debe dinero decide secuestrar al deudor para obtener el dinero del rescate por parte de su padre.
Decide-kidnap-lovers	Un personaje al cual se le debe dinero decide secuestrar al deudor para obtener el dinero del rescate por parte de su amante.
Kidnap	Un personaje secuestra a otro.
Pay-rescue-father	Un personaje paga el rescate de su hijo secuestrado.
Pay-rescue-lover	Un personaje paga el rescate de su amante secuestrado
Decide-rob-lovers	Un personaje amante de otro secuestrado decide obtener dinero para pagar el rescate.
Decide-rob-father-son	Un personaje padre de otro secuestrado decide obtener dinero para pagar el rescate.
Rob-business	Un personaje armado roba en el negocio de otro.
Rob-bank	Un personaje armado roba en un banco.
No-motivation-money	Un personaje con motivación de obtener dinero deja de tener esta motivación al alcanzar el máximo nivel.

Tabla 20. Acciones de dinero

5.3.4 Acciones de la policía

La inclusión de la policía como una serie de personajes con motivaciones contrarias al resto añade complejidad a los planes obtenidos, ya que en muchas ocasiones obstaculizarán los objetivos de los personajes que cometen actos delictivos. Los personajes que son policías empezarán a sospechar de un individuo si ha cometido un asesinato o ha robado un banco. Cuando un policía está sospechando de otro personaje, puede ejecutarse la acción *follow*, de manera que sigue a este segundo personaje. Debido a que alguien que ha cometido un crimen posiblemente deje el arma que ha utilizado en algún lugar posibilita al policía para que la recoja y obtenga una prueba delictiva, pudiendo en este momento capturar al criminal. Sin embargo, una persona en la cárcel puede escaparse, por lo que puede seguir teniendo influencia en el plan. La siguiente acción es la que permite capturar a un criminal.

```
(:action capture
  :parameters (?police ?person - person ?place - genericplace
    ?jail - jail ?vengeance - vengeance)
  :precondition (and
    (not(non-movable ?police))
    (not(non-movable ?person))
    (at ?police ?place)
    (at ?person ?place)
    (intends-to-capture ?police ?person))
  :effect (and
    (not(at ?person ?place))
    (at ?person ?jail)
    (non-movable ?person)
    (not(suspecting-about ?police ?person))
    (not(intends-to-capture ?police ?person))
    (has-motive ?person ?police ?vengeance)))
```

Es preciso destacar que el policía debe estar en la misma posición que el criminal, y que como efecto de la acción se genera un motivo de venganza del criminal para con el policía.

Acción	Descripción
Find-clue-object	Un policía descubre una prueba delictiva que incrimina a otro personaje y pretende capturarlo.
Suspect-police	Un policía sospecha que un personaje ha matado a otro.
Follow	Un policía que sospecha de otro personaje lo sigue desde su posición hasta donde está este último.
Capture	Un policía captura y encarcela a otro personaje.
Escape-from-jail	Un personaje encarcelado escapa de la cárcel.

Tabla 21. Acciones de policía

5.3.5 Acciones emocionales

Se han incluido también acciones referentes a las relaciones personales, de manera que podrán influir también en la historia como un elemento más. De esta forma, un personaje hombre podrá besar a un personaje mujer, de manera que se convierten en amantes. Si uno de los personajes que se besan ya es amante de un tercero, éste lo descubrirá y obtendrá un motivo de asesinato de celos. Esta última acción explicada se muestra a continuación:

```
(:action discover-kissing
  :parameters      (?p1 ?p3 - male ?p2 - female ?place -
genericplace ?jealousy - jealousy)
  :precondition    (and
                    (not (non-movable ?p1))
                    (not (non-movable ?p2))
                    (not (non-movable ?p3))
                    (lovers ?p1 ?p2)
                    (kissed ?p3 ?p2)
                    (not (= ?p1 ?p3)))
  :effect          (and
                    (has-motive ?p1 ?p2 ?jealousy)
                    (has-motive ?p1 ?p3 ?jealousy)))
```

Acción	Descripción
Kiss	Un personaje masculino al que le gusta uno femenino lo besa.
Discover-kissing	Un personaje masculino que tiene una amante, la cual ha besado a otro personaje, adquiere motivos de asesinato sobre estos dos personajes.

Tabla 22. Acciones emocionales

5.3.6 Acciones relacionadas con las organizaciones criminales

Existen ciertas acciones relacionadas con la estructura de las organizaciones criminales modeladas en el dominio. En primer lugar, se contempla la posibilidad de que el líder de una organización muera, pasando el liderazgo a su sucesor, si existe. También es posible que un miembro de una organización deje de serlo, generando un motivo de asesinato por traición por parte del líder.

Por último, una deuda por parte de un dueño de un negocio con un líder de una organización puede desembocar en la destrucción del negocio por este último. Se generará entonces un motivo de venganza hacia el jefe de la organización. La acción comentada se muestra a continuación.

```
(:action destroy-business
  :parameters (?bus - business ?or - organization ?boss ?person
- person ?vengeance - vengeance)
  :precondition (and
    (not(non-movable ?boss))
    (not(non-movable ?person))
    (has-business ?person ?bus)
    (debt ?person ?boss)
    (boss-organization ?boss ?or)
    (not(destroyed ?bus)))
  :effect (and
    (destroyed ?bus)
    (has-motive ?person ?boss ?vengeance)))
```

Acción	Descripción
New-boss	Al morir el líder de una organización, su sucesor pasa a ser el nuevo líder.
Betray	Un miembro de una organización deja de estar en la misma y el líder adquiere un motivo de asesinato de traición sobre el primero.
Betray-bodyguard	Un miembro de una organización y guardaespaldas del líder de la misma deja de estar en la organización y de ser guardaespaldas. El líder adquiere un motivo de asesinato de traición sobre el personaje.
Destroy-business	Una organización destruye el negocio de un personaje que tiene una deuda con el líder de la organización.

Tabla 23. Acciones de organizaciones criminales

5.4 DEFINICIÓN DE PROBLEMAS

Es posible generar nuevos problemas a medida para obtener historias diferentes. Sin embargo, existe una serie de restricciones a la hora de crear los problemas. A continuación se explicarán las normas de creación de problemas.

- Los problemas estarán escritos en PDDL1.2 y llevarán la extensión *.pddl*.
- Será obligatorio definir una sola instancia de los siguientes tipos:
 - Jealousy.
 - Treason
 - Vengeance
 - Nomoney

- Lowmoney
 - Mediummoney
 - Highmoney
- Será recomendable si se desea que se ejecuten el mayor número de acciones diferentes que se defina al menos una instancia de los siguientes tipos:
 - Male
 - Female
 - Genericplace
 - Business
 - Jail
 - Kidnapplace
 - Bank
 - Policestation
 - Weapon
 - Organization
- Será obligatorio definir la jerarquía de niveles monetarios con los siguientes predicados:
 - *(next-money-level nomoney lowmoney)*
 - *(next-money-level lowmoney mediummoney)*
 - *(next-money-level mediummoney highmoney)*
- Será obligatorio definir el nivel monetario de cada personaje mediante un predicado por cada personaje del tipo: *(has-money-level person mediummoney)*
- Será obligatorio definir la posición inicial de cada personaje mediante un predicado por cada uno del tipo: *(at person place)*
- Será obligatorio definir la posición inicial de cada una de las armas instanciadas, mediante predicados del tipo *(at weapon place)* o *(has-object person weapon)*.

De esta forma, los problemas tendrán la siguiente estructura:

```
(define (problem problem1) (:domain Policiaco)
(objects
  male1 male2 ... maleN - male
  female1 female2 ... femaleN - female
  place1 place2... placeN - genericplace
  business1 business2 ... businessN -business
  jaill1 jail2 ... jailN - jail
  policestation1 policestation2 ... policestationN - policestation
  bank1 bank2 ... bankN - bank
  kidnapplace1 kidnapplace2 ... kidnapplaceN - kidnapplace
  weapon1 weapon2 ... weaponN - weapon
  organization1 organization2 ... organizationN - organization
  jealousy - jealousy
  treason - treason
  vengeance - vengeance
  nomoney - nomoney
  lowmoney - lowmoney
  mediummoney - mediummoney
  highmoney - highmoney
)
```

```
(:init
  (at male1 place1)
  (at male2 place2)
  ...
  (at weapon1 place1)
  ...
  (has-money-level male1 lowmoney)
  ...
  (next-money-level nomoney lowmoney)
  (next-money-level lowmoney mediummoney)
  (next-money-level mediummoney highmoney)

  ;Resto de predicados
  ...
)
(:goal (and
  (...)
))
)
```

A continuación se muestra un ejemplo de problema para este dominio, problem1.pddl, que es utilizado en la fase de experimentación:

```
(define (problem policiaco1) (:domain Policiaco)
(objects
  vito michael lucabراس alneri sollozzo captain-mccluskey captain-
oneill louis nino barzini - male
restaurant-louis jewelry-store-nino - business
kay amy marie - female
gun machine-gun knife shotgun revolver rifle - weapon
corleone-family barzini-family - organization
jealousy - jealousy
treason - treason
vengeance - vengeance
house-corleone house-alneri house-lucabراس house-sollozzo house-
barzini house-amy market train-station casino - genericplace
kidnapplace - kidnapplace
jail - jail
policestation - policestation
bank - bank
nomoney - nomoney
lowmoney - lowmoney
mediummoney - mediummoney
highmoney - highmoney
)

(:init
;Fijas-----
  (next-money-level nomoney lowmoney)
  (next-money-level lowmoney mediummoney)
  (next-money-level mediummoney highmoney)

  (is-police captain-mccluskey)
  (is-police captain-oneill)

  (has-business louis restaurant-louis)
```



```

(has-business nino jewelry-store-nino)

(boss-organization barzini barzini-family)
(boss-organization vito corleone-family)
(successor-organization michael corleone-family)
(is-boss vito)
(is-boss barzini)
(in-organization sollozzo barzini-family)
(in-organization michael corleone-family)
(in-organization alneri corleone-family)
(in-organization lucabراسي corleone-family)

(lovers michael kay)
(lovers nino marie)
(father-son vito michael)

;Obligatorias y modificables-----
(at vito house-corleone)
(at michael house-corleone)
(at lucabراسي house-lucabراسي)
(at alneri house-alneri)
(at sollozzo market)
(at barzini house-barzini)
(at captain-mccluskey policestation)
(at captain-oneill train-station)
(at louis restaurant-louis)
(at marie jewelry-store-nino)
(at nino jewelry-store-nino)
(at amy house-amy)

(at gun restaurant-louis)
(at machine-gun house-corleone)
(at knife market)
(at shotgun house-sollozzo)
(at revolver house-lucabراسي)
(at rifle house-barzini)

(has-money-level vito mediummoney)
(has-money-level michael highmoney)
(has-money-level lucabراسي lowmoney)
(has-money-level alneri nomoney)
(has-money-level sollozzo nomoney)
(has-money-level barzini mediummoney)
(has-money-level kay nomoney)
(has-money-level captain-mccluskey mediummoney)
(has-money-level captain-oneill nomoney)
(has-money-level louis highmoney)
(has-money-level marie lowmoney)
(has-money-level nino highmoney)
(has-money-level amy mediummoney)

;Añadibles-----
(motivation-money lucabراسي)
(motivation-money sollozzo)
)
(:goal
  (dead lucabراسي)))

```

5.5 CREACIÓN DE FRASES

El planificador Sayphi, utilizando el dominio y un problema, podrá crear planes formados por una sucesión de acciones. Sin embargo, para que el resultado de la ejecución sea una narrativa se utiliza la función *narrate* del planificador. En esta función, al generarse un plan se genera un texto por cada una de las acciones de las que consta.

Por ejemplo, para la acción *kill*, que se mostraría de la siguiente manera si no se utiliza la función *narrate*: (KILL LUCABRASI MICHAEL HOUSE-CORLEONE GUN). Se ha definido el texto: LUCABRASI kills MICHAEL with GUN at HOUSE-CORLEONE. Todas las acciones del dominio serán representadas por una frase en caso de aparecer en un plan.

En la siguiente tabla se muestra un ejemplo de una frase para cada acción del dominio.

Acción	Ejemplo
Move-to-kill	Michael goes from House-Corleone to House-Barzini to kill Barzini.
Move-to-kill-by-command	Michael goes from House-Corleone to House-Barzini to kill Barzini by Vito's command.
Move-to-kill-bodyguard	Michael goes from House-Corleone to House-Barzini to kill Sollozzo, Barzini's bodyguard.
Move-to-kill-bodyguard-by-command	Michael goes from House-Corleone to House-Barzini to kill Sollozzo, Barzini's bodyguard, by Vito's command.
Move-to-kidnap	Michael goes from House-Corleone to House-Amy to kidnap Amy.
Move-to-rob-bank	Michael goes from House-Corleone to bank to commit a bank robbery.
Move-to-rob-business	Michael goes from House-Corleone to Restaurant-Louis, Louis' business, to commit a robbery.
Move-to-kiss	Michael goes from House-Corleone to Market to kiss Kay.
Move-to-pickup-object-kill	Michael goes from House-Corleone to Jewelry-store-Nino and picks up gun to kill Sollozzo
Move-to-pickup-object-kill-by-command	Michael goes from House-Corleone to Market and picks up Knife to kill Sollozzo by Vito's command.
Move-to-pickup-object-rob	Michael goes from House-Corleone to Casino and picks up gun to rob.
Pick-up-object-to-kill	Michael picks up gun at House-Corleone to kill Sollozzo.
Pick-up-object-to-kill-by-command	Michael picks up gun at House-Corleone to kill Sollozzo by Vito's command.
Pick-up-object-to-rob	Michael picks up gun at House-Corleone to rob.
Drop-object	Michael drops gun at House-Corleone.
Decide-murder	Michael decide to kill Sollozzo for jealousy.
Command-murder	Vito command Michael to kill Luca Brasi for treason.
Kill	Michael kills Sollozzo with gun at Casino.

Kill-by-command	Michael kills Sollozzo with gun at Casino by Vito's command.
Kill-bodyguard	Michael kills Sollozzo, Barzini's bodyguard, with gun at Casino.
Kill-bodyguard-by-command	Michael kills Sollozzo, Barzini's bodyguard, with gun at Casino.
Failed-kill	Michael tries to kill Sollozzo at Casino, but fails. Sollozzo gets hurt.
Failed-kill-by-command	Michael tries to kill Sollozzo at Casino by Vito's command, but fails. Sollozzo gets hurt.
Vengeance-lover	Michael wants to get revenge on Sollozzo for killing his lover Kay.
Vengeance-father	Michael wants to get revenge on Sollozzo for killing his father Vito.
Kiss	Michael kisses Amy at House-Amy.
Discover-kissing	Michael discovers that Sollozzo kissed his lover Kay and wants to get revenge on them.
Find-clue-object	Captain McCluskey finds at train-station an evidence on knife that incriminates Michael.
Suspect-police	Captain McCluskey suspects that Michael killed Sollozzo.
Follow	Michael silently chases Michael from Restaurant-Louis to Market.
Capture	Michael captures Michael at Casino and sends him to jail.
Escape-from-jail	Michael escapes from jail and goes to House-Corleone.
New-boss	Michael becomes new boss of Corleone-Family.
Betray	Michael betrays Vito and quits Corleone-Family.
Betray-bodyguard	Luca Brasi, Vito's bodyguard, betrays him and quits Corleone-Family.
Destroy-business	Corleone-Family destroys Restaurant-Louis, Louis' business, due to an unpaid debt with Vito.
Loan	Michael lends money to Al Neri.
Pay-debt	Al Neri pays off the debt with Michael.
Decide-kidnap-father-son	Michael decide to kidnap Al Neri due to an unpaid debt.
Decide-kidnap-lovers	Michael decide to kidnap Al Neri due to an unpaid debt.
Kidnap	Michael attacks Amy at House-Amy. Amy gets kidnapped by Michael.
Pay-rescue-father	Vito pays Barzini for the release of his son Michael.
Pay-rescue-lover	Kay pays Barzini for the release of lover Michael.
Decide-rob-lovers	Michael decide to rob in order to make money to pay Barzini for the release of lover Kay.
Decide-rob-father-son	Vito decide to rob in order to make money to pay Barzini for the release of son Michael.
Rob-business	Michael robs at Restaurant-Louis, Louis' business.
Rob-bank	Michael commits a bank robbery at bank.
No-motivation-money	Michael has a lot of money and won't rob in the future.

Tabla 24. Frases de acciones

6. EXPERIMENTACIÓN

El análisis del funcionamiento del sistema de planificación con el dominio desarrollado constituye una de las partes más importantes de este trabajo. El interés narrativo de las historias que se obtengan y la variabilidad entre las mismas permitirá determinar los puntos fuertes y las carencias del dominio, así como los aciertos de diseño y la adecuación del planificador utilizado. Será importante que el planificador resuelva la mayor cantidad de problemas posible y que soporte también gran cantidad de elementos en la historia. Además, como ya se ha comentado anteriormente, se valorará que el planificador obtenga distintas historias con el mayor índice de variabilidad posible para el mismo problema y la misma configuración de ejecución.

Para llevar a cabo estas pruebas se generarán una serie de problemas en los que variarán el estado inicial y el estado meta. Se ejecutará el planificador utilizando configuraciones de aleatoriedad distintas repetidas veces para un mismo problema. Se observará qué efecto tiene sobre la longitud, la variabilidad y el interés narrativo de los planes aumentar la aleatoriedad de la planificación. También se observará si la inclusión de más información en el estado inicial y en el final afecta a la generación de planes y cómo lo hace.

6.1 PROBLEMAS UTILIZADOS

Para la fase de experimentación se generarán distintos problemas de diferente longitud y complejidad. La base de los problemas será un escenario con un conjunto de objetos basados en la película “El Padrino” (Francis Ford Coppola, 1972). De esta forma, para la gran mayoría de problemas se mantendrán los mismos objetos, añadiéndose puntualmente otros nuevos. Asimismo, habrá una serie de predicados que se mantendrán fijos en todos los problemas.

6.1.1 Elementos fijos: objetos

- Personajes masculinos:
 - Vito
 - Michael
 - Luca Brasi
 - Al Neri
 - Sollozzo
 - Captain McCluskey
 - Captain O’Neill
 - Louis
 - Nino
 - Barzini
- Personajes femeninos:
 - Kay
 - Amy
 - Marie

- Armas:
 - Gun
 - Machine-Gun
 - Knife
 - Shotgun
 - Revolver
 - RIFLE
- Lugares
 - House Corleone
 - House Al Neri
 - House Luca Brasi
 - House Sollozzo
 - House Barzini
 - House e Amy
 - Market
 - Train-Station
 - Casino
- Negocios
 - Restaurant-Louis
 - Jewelry-store-Nino
- Jail
- Bank
- Policestation
- Organizaciones criminales
 - Familia Corleone
 - Familia Barzini

6.1.2 Elementos fijos: predicados

Debido a que los personajes de Captain McCluskey y Captain O'Neill están pensados para ser policías, y siendo recomendable que haya en la historia más de un agente de la ley, serán fijos los predicados:

```
(is-police captain-mccluskey)
(is-police captain-oneill)
```

Será también constante la información respecto a las organizaciones criminales. Existen dos organizaciones criminales en la historia: la Familia Corleone y la Familia Barzini. Vito es el líder de la Familia Corleone, y Michael, además de ser un miembro de la misma, es el sucesor al mando. También son miembros de esta organización Luca Brasi y Al Neri. Respecto a la Familia Barzini, Barzini es el líder. En este caso no hay ningún sucesor, pero existe un miembro, Sollozzo. Los personajes de Louis, Nino, Amy, Kay y Marie no pertenecen a ninguna organización. Esta información se modela tal cual se muestra a continuación:

```
(boss-organization barzini barzini-family)
(boss-organization vito corleone-family)
(successor-organization michael corleone-family)
(is-boss vito)
(is-boss barzini)
(in-organization sollozzo barzini-family)
(in-organization michael corleone-family)
(in-organization alneri corleone-family)
(in-organization lucabراسi corleone-family)
```

Además, en todos los problemas Nino y Louis tendrán sus respectivos negocios:

```
(has-business louis restaurant-louis)
(has-business nino jewelry-store-nino)
```

Finalmente, como último elemento fijo, se determina que Vito es el padre de Michael y que este último y Kay son amantes mediante los predicados:

```
(father-son vito michael)
(lovers michael kay)
```

6.1.3 Elementos variables

Los elementos que variarán en los problemas con los que se lleve a cabo la experimentación son numerosos. En primer lugar, las posiciones iniciales de personajes y objetos pueden afectar ostensiblemente a los planes generados: que los personajes tengan que realizar más viajes para recoger objetos y cumplir sus objetivos hará que se generen planes más largos y variados. El dinero que tiene cada personaje es otro de los elementos que variarán entre los problemas. En un problema donde el dinero que posean los personajes sea escaso en general habrá más robos, y en aquellos en los que los niveles de dinero estén muy descompensados habrá muchos préstamos. Además, se probará que distintos personajes tengan la motivación de conseguir dinero, y se estudiará qué ocurre cuando la gran mayoría tienen esta motivación. Para completar los predicados asociados con el dinero, pese a que se pueden generar espontáneamente, se incluirán deudas entre personajes desde el principio en el estado final de algunos problemas y se observará su efecto.

También es interesante añadir predicados referentes a las relaciones personales, como *likes* y *lovers*, de manera que observemos si añaden complejidad a los planes generados, y si éstos tienen un mayor componente de violencia. También se estudiará cómo la inclusión de guardaespaldas afecta a las historias generadas.

Se crearán también problemas en los cuales haya habido eventos anteriormente, de manera que se podrán incluir en el estado inicial predicados tales como *has-motive*, *intends-to-kill* o *kidnapped-by*.

En los problemas de prueba creados para esta fase de experimentación se han variado los predicados mostrados en la siguiente tabla.

Predicado	Observaciones
At (personas)	Obligatorio, se modifican los parámetros.
At (armas)	Obligatorio, se modifican los parámetros.
Has-money-level	Obligatorio, se modifican los parámetros.
Motivation-money	No obligatorio.
Likes	No obligatorio.
Debt	No obligatorio.
Has-motive	No obligatorio.
Has-object	No obligatorio. Sustituye al predicado <i>at</i> referente al objeto referenciado en <i>has-object</i> .
Kidnapped-by	No obligatorio.
Non-movable	No obligatorio. Debe activarse junto con <i>kidnapped-by</i> , <i>dead</i> o (<i>at person jail</i>)
Bodyguard	No obligatorio.
Has-bodyguard	No obligatorio. Debe activarse junto con <i>bodyguard</i> .

Tabla 25. Predicados variables en los problemas.

6.1.4 Metas

Pese a que los predicados incluidos en el estado inicial influirán en el desarrollo de las historias generadas, son las metas los elementos que permiten al usuario que genera la historia decidir acerca de la misma. De esta forma, durante la fase de experimentación se comprobará cómo añadiendo únicamente una meta es posible cambiar significativamente las historias que se generan.

Se comenzará creando problemas con un estado meta simple, con uno o dos predicados únicamente, y se irán incluyendo más paulatinamente. Además, se crearán problemas en los que los predicados del estado final sean independientes, por ejemplo: que se haya producido un asesinato de un personaje, que otro haya obtenido el máximo nivel de dinero, y que dos personajes sean ahora amantes. Se observará cómo la cantidad de predicados en el estado meta influye en la complejidad y el interés narrativo de las historias generadas.

6.2 PLAN DE PRUEBAS

Se ejecutará el planificador para cada uno de los programas de prueba utilizando tres valores distintos de aleatoriedad (parámetro *:probability-ehc*): 0.2, 0.5 y 0.8. Se utilizará el parámetro *:number-plans* para generar 5 planes para cada configuración y observar la variabilidad entre los mismos dependiendo de la aleatoriedad. El tiempo consumido por cada ejecución puede ser considerable si el problema tiene cierta complejidad, por lo que se ha fijado un *timeout* de 300 segundos.

Para cada problema se generará una tabla en la que se incluye información cuantitativa acerca de los resultados obtenidos. Para cada ejecución se mostrará si se ha generado un plan satisfactorio, la longitud del mismo, la diversidad del plan con respecto al anterior y

el número de acciones distintas que tienen lugar en el plan. La diversidad del plan con respecto al plan anterior se medirá como el número de acciones que aparecen en uno de los dos planes pero no en el otro, teniendo en cuenta que en este caso solo se tomarán como acciones iguales aquellas en las que coincide la acción y todos los parámetros. Por otra parte, para el número de acciones distintas mostrado en la tabla las acciones se contabilizarán desde el punto de vista narrativo y no desde el punto de vista de planificación (se contabilizarán todas las acciones de *move* como una sola, y de *kill* como una sola), pues esta medida posibilitará evaluar la calidad narrativa de los planes.

Los datos de los planes no satisfactorios no se utilizarán para calcular la media de los valores para cada problema. Sin embargo, estos planes que no alcanzan todas las metas pueden ser muy interesantes, como se comentará en los siguientes apartados.

6.3 PROBLEMA 1

6.3.1 Definición

Como comienzo de la fase de experimentación se ha utilizado un problema sencillo. Éste cuenta básicamente con los elementos obligatorios comentados anteriormente: la estructura de las organizaciones, los roles de los personajes y las relaciones afectivas son las mismas. Los personajes y las armas se encuentran repartidos en los distintos lugares geográficos, y los niveles de dinero son dispares. Cabe destacar que los personajes de Luca Brasi y de Sollozzo, cada uno de una organización criminal, comienzan con niveles mínimos de dinero, y que ambos tienen la motivación de obtenerlo ilegalmente. Además, no se ha incluido ninguna motivación de asesinato ni ningún predicado *likes*.

Se establece que Luca Brasi debe estar muerto al acabar el plan.

6.3.2 Resultados

Para el problema más sencillo, el planificador encuentra solución en todas las ocasiones en que ejecuta. Debido a que solo hay una condición en el estado meta, los planes obtenidos son cortos y muy similares entre sí. Asimismo, las diferencias entre la ejecución con el valor de aleatoriedad más alto y el más bajo son muy pequeñas, siendo lo único a destacar que los planes son menos aleatorios. Esto se debe a que se escoge en cada instante la mejor acción para llegar al estado final.

Con pequeñas variaciones, el tipo de plan que se obtiene para este problema es el siguiente, obtenido con el valor menos aleatorio:

```
LUCABRASI goes from HOUSE-LUCABRASI to HOUSE-BARZINI and picks up RIFLE to
rob
LUCABRASI betrays VITO and quits CORLEONE-FAMILY
VITO command MICHAEL to kill LUCABRASI for TREASON
MICHAEL goes from HOUSE-CORLEONE to HOUSE-LUCABRASI and picks up
REVOLVER to kill LUCABRASI by VITO's command
MICHAEL goes from HOUSE-LUCABRASI to HOUSE-BARZINI to kill LUCABRASI by
VITO's command
MICHAEL kills LUCABRASI with REVOLVER at HOUSE-BARZINI by VITO's command
```

Los planes más aleatorios incluyen una acción en la que Luca Brasi tira su arma justo tras traicionar a su líder, de manera que no la ha utilizado aún. Por ello, y debido a que el

factor de variabilidad es similar independientemente del valor de *probability-ehc* escogido, los planes menos aleatorios tienen mayor interés.

Aleatoriedad alta						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	8	8	7	8	8	7,8
Diversidad	-	0	9	9	0	4,5
Nº acciones distintas	6	6	6	6	6	6
Aleatoriedad media						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	7	8	6	8	6	7
Diversidad	-	9	8	8	8	8,25
Nº acciones distintas	6	6	5	6	5	5,6
Aleatoriedad baja						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	7	6	8	6	6	6,6
Diversidad	-	7	4	8	8	6,75
Nº acciones distintas	6	5	6	5	6	5,6

Tabla 26. Datos problema 1

6.4 PROBLEMA 2

6.4.1 Definición

En esta segunda prueba se ha utilizado como base el problema 1, al cual se le ha añadido un cambio al estado meta. Se determina que al final del plan, aparte de que Luca Brasi esté muerto, Sollozzo debe haber alcanzado el máximo nivel de dinero. Recordemos que éste tiene la motivación de conseguir dinero. De esta forma, haremos que estos dos personajes pasen a ser protagonistas de las historias resultantes de este problema, que a priori serán más largas y complejas que las del anterior problema.

Las posiciones de los personajes y las armas siguen siendo las mismas, y se observará cómo éstas influyen en el desarrollo de planes más largos.

6.4.2 Resultados

Existe una diferencia notable de complejidad y de longitud de los planes obtenidos, teniendo en cuenta que la única modificación es la inclusión en el estado meta de que el personaje de Sollozzo debe acabar con el máximo nivel de dinero. De esta forma, mientras que con el primer problema la historia se centraba en el personaje de Luca Brasi y su relación con su organización, en los planes generados con este problema se incluye una historia paralela con Sollozzo como protagonista. Como ocurría con el anterior problema, los planes obtenidos no varían en exceso. Los cambios más importantes son el orden en que ocurren los sucesos, los lugares que visitan los personajes y las armas que utilizan. Sí

cabe destacar que estos cambios son más frecuentes con la planificación más aleatoria. Además, se han generado planes satisfactorios en todas las ocasiones.

El plan más interesante de los obtenidos para este problema es el siguiente.

SOLLOZZO picks up KNIFE at MARKET to rob
 SOLLOZZO goes from MARKET to RESTAURANT-LOUIS , LOUIS's business, to commit a robbery
 LUCABRASI goes from HOUSE-LUCABRASI to HOUSE-BARZINI and picks up RIFLE to rob
 MICHAEL lends money to SOLLOZZO
 SOLLOZZO robs at RESTAURANT-LOUIS, LOUIS's business
 SOLLOZZO goes from RESTAURANT-LOUIS to JEWELRY-STORE-NINO , NINO's business, to commit a robbery
 SOLLOZZO robs at JEWELRY-STORE-NINO, NINO's business
 LUCABRASI betrays VITO and quits CORLEONE-FAMILY
 VITO command MICHAEL to kill LUCABRASI for TREASON
 MICHAEL goes from HOUSE-CORLEONE to HOUSE-LUCABRASI and picks up REVOLVER to kill LUCABRASI by VITO's command
 MICHAEL goes from HOUSE-LUCABRASI to HOUSE-BARZINI to kill LUCABRASI by VITO's command
 MICHAEL kills LUCABRASI with REVOLVER at HOUSE-BARZINI by VITO's command

Aleatoriedad alta						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	13	11	15	12	12	12,6
Diversidad	-	10	14	19	12	13,75
Nº acciones distintas	8	8	8	8	8	8
Aleatoriedad media						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	13	13	14	14	12	13,2
Diversidad	-	12	9	0	12	8,25
Nº acciones distintas	8	8	9	9	8	8,4
Aleatoriedad baja						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	12	12	14	12	13	12,6
Diversidad	-	8	4	4	7	5,75
Nº acciones distintas	8	8	8	8	8	8

Tabla 27. Datos problema 2

6.5 PROBLEMA3

6.5.1 Definición

En esta tercera prueba se ha utilizado de nuevo el problema de la anterior, al cual se le han aplicado algunos cambios. En primer lugar, se ha sustituido la motivación de que Luca Brasi consiga el máximo nivel de dinero por que lo haga Michael. Además, se han incluido los predicados siguientes para determinar que Luca Brasi y Sollozzo se sienten atraídos por Amy.

```
(likes lucabراسي amy)
(likes sollozzo amy)
```

Por otra parte, las condiciones finales han sido modificadas. Se pretende generar historias en los que Sollozzo muera, Al Neri acabe en la cárcel y Luca Brasi haya besado a Amy. A priori, el problema no tiene grandes diferencias con los anteriores. Sin embargo, al incluirse predicados muy diferentes e independientes entre sí en las condiciones finales se espera que se ejecuten más acciones en los planes y que éstos sean más variados entre sí. Además, la inclusión del encarcelamiento de un personaje hace que miembros de la policía sean incluidos en los planes.

6.5.2 Resultados

Tras la realización de las pruebas lo primero que cabe destacar es que la ejecución con alta aleatoriedad genera planes muy largos pero en los que ocurren muchas acciones que no aportan nada a las historias. Asimismo, como cabría esperar, los planes generados con este índice de aleatoriedad son muy diferentes entre sí.

Respecto a las narrativas generadas con aleatoriedad media y alta, las diferencias no son muy notables. Los planes obtenidos difieren entre sí, pero el interés narrativo de los mismos es semejante. Es necesario destacar que las historias generadas son ya complejas e incluyen una gran cantidad de personajes y acciones distintos. Además, con los índices de aleatoriedad medio y alto se evitan acciones innecesarias que restan realismo a las historias. La narrativa más interesante de entre las obtenidas es la siguiente:

```
MICHAEL picks up MACHINE-GUN at HOUSE-CORLEONE to rob
MICHAEL goes from HOUSE-CORLEONE to BANK to commit a bank robbery
SOLLOZZO goes from MARKET to HOUSE-BARZINI and picks up RIFLE to rob
SOLLOZZO betrays BARZINI and quits BARZINI-FAMILY
SOLLOZZO goes from HOUSE-BARZINI to RESTAURANT-LOUIS , LOUIS's business,
to commit a robbery
MICHAEL commits a bank robbery at BANK
MICHAEL goes from BANK to RESTAURANT-LOUIS , LOUIS's business, to commit a
robbery
LUCABRASI goes from HOUSE-LUCABRASI to HOUSE-AMY to kiss AMY
CAPTAIN-MCCLUSKEY silently chases MICHAEL from POLICESTATION to
RESTAURANT-LOUIS
LUCABRASI kisses AMY at HOUSE-AMY
MICHAEL betrays VITO and quits CORLEONE-FAMILY
VITO command ALNERI to kill MICHAEL for TREASON
SOLLOZZO robs at RESTAURANT-LOUIS, LOUIS's business
SOLLOZZO drops RIFLE at RESTAURANT-LOUIS
ALNERI goes from HOUSE-ALNERI to RESTAURANT-LOUIS and picks up GUN to kill
MICHAEL by VITO's command
```

LOUIS decide to kill SOLLOZZO for VENGEANCE
 ALNERI kills MICHAEL with GUN at RESTAURANT-LOUIS by VITO's command
 LOUIS picks up RIFLE at RESTAURANT-LOUIS to kill SOLLOZZO
 ALNERI drops GUN at RESTAURANT-LOUIS
 LOUIS kills SOLLOZZO with RIFLE at RESTAURANT-LOUIS
 CAPTAIN-MCCLUSKEY finds at RESTAURANT-LOUIS an evidence on GUN that incriminates ALNERI
 CAPTAIN-MCCLUSKEY captures ALNERI at RESTAURANT-LOUIS and sends him to JAIL

Aleatoriedad alta						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	44	38	40	35	52	41,8
Diversidad	-	59	52	43	42	49
Nº acciones distintas	12	14	14	14	14	13,6
Aleatoriedad media						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	23	44	22	19	38	29,2
Diversidad	-	48	50	23	37	39,5
Nº acciones distintas	13	14	13	14	13	13,4
Aleatoriedad baja						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	23	33	38	24	32	30
Diversidad	-	32	33	40	35	35
Nº acciones distintas	13	15	14	13	14	13,8

Tabla 28. Datos problema 3

6.6 PROBLEMA 4

6.6.1 Definición

Mediante este problema se probarán la correcta ejecución de acciones que no han tenido lugar en los planes comentados anteriormente. En concreto, se pretende que haya un secuestro, para lo cual se ha incluido como meta, y debe haber un cambio de líder en la Familia Corleone. Además, Luca Brasi deberá alcanzar el máximo nivel de dinero y debe estar en la cárcel al final del plan.

Para obtener planes interesantes se han añadido en el estado inicial dos deudas: de Kay a Barzini y de Louis a Vito. Además, se ha determinado que desde el principio Michael tiene un motivo de asesinato para con Sollozzo.

6.6.2 Resultados

Las narrativas obtenidas con este problema son muy variadas. Las acciones ejecutadas y los objetos involucrados en las mismas han variado ostensiblemente entre los distintos planes, incluyendo aquellos generados con baja aleatoriedad. Los planes obtenidos son interesantes desde el punto de vista narrativo, ya que en ellos tienen lugar una gran cantidad de acciones distintas y la interacción entre los personajes es muy alta. Sin embargo, para este caso obviaremos los planes obtenidos con aleatoriedad alta, pues en ellos tienen lugar acciones innecesarias que restan realismo a las historias.

De entre los planes generados con aleatoriedad media y alta, el más interesante es el siguiente:

LUCABRASI betrays VITO and quits CORLEONE-FAMILY
BARZINI decide to kidnap KAY due to an unpaid debt
BARZINI goes from HOUSE-BARZINI to HOUSE-CORLEONE to kidnap KAY
BARZINI attacks KAY at HOUSE-CORLEONE. KAY gets kidnapped by BARZINI
LUCABRASI goes from HOUSE-LUCABRASI to HOUSE-CORLEONE and picks up MACHINE-GUN to rob
MICHAEL decide to kill SOLLOZZO for VENGEANCE
LUCABRASI goes from HOUSE-CORLEONE to BANK to commit a bank robbery
VITO goes from HOUSE-CORLEONE to JEWELRY-STORE-NINO to kiss MARIE
LUCABRASI commits a bank robbery at BANK
LUCABRASI goes from BANK to RESTAURANT-LOUIS , LOUIS's business, to commit a robbery
MICHAEL goes from HOUSE-CORLEONE to RESTAURANT-LOUIS and picks up GUN to kill SOLLOZZO
VITO command MICHAEL to kill LUCABRASI for TREASON
MICHAEL tries to kill LUCABRASI at RESTAURANT-LOUIS by VITO's command, but fails.
LUCABRASI gets hurt
LUCABRASI goes from RESTAURANT-LOUIS to JEWELRY-STORE-NINO , NINO's business, to commit a robbery
LUCABRASI decide to kill VITO for VENGEANCE
LUCABRASI kills VITO with MACHINE-GUN at JEWELRY-STORE-NINO
CAPTAIN-ONEILL silently chases LUCABRASI from TRAIN-STATION to JEWELRY-STORE-NINO
LUCABRASI robs at JEWELRY-STORE-NINO, NINO's business
LUCABRASI drops MACHINE-GUN at JEWELRY-STORE-NINO
CAPTAIN-ONEILL finds at JEWELRY-STORE-NINO an evidence on MACHINE-GUN that incriminates LUCABRASI
CAPTAIN-ONEILL captures LUCABRASI at JEWELRY-STORE-NINO and sends him to JAIL
MICHAEL becomes new boss of CORLEONE-FAMILY

Aleatoriedad alta						
	1	2	3	4	5	Media
Genera plan	Plan parcial	Sí	Sí	Sí	Sí	
Longitud	19	37	28	34	23	30,5
Diversidad	-	41	55	40	51	46,75
Nº acciones distintas	9	17	14	17	14	15,5
Aleatoriedad media						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	24	28	29	28	24	26,6
Diversidad	-	42	41	45	38	41,5
Nº acciones distintas	16	15	14	14	16	15
Aleatoriedad baja						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	26	28	22	24	23	24,6
Diversidad	-	44	42	24	32	35,5
Nº acciones distintas	16	14	16	16	16	15,6

Tabla 29. Datos problema 4

6.7 PROBLEMA 5

6.7.1 Definición

Mediante este problema, con una complejidad similar al anterior, se pretende obtener planes con mayor realismo y menor número de acciones innecesarias. Para ello, se ha incluido un motivo de asesino de Sollozzo para con Vito, además de que el primero ya posee un arma en su poder. Además, se incluyen tres predicados *likes*, de manera que pese a que no hay objetivos de carácter afectivo en el estado meta, influirán en la historia. Además, ahora son los personajes de Al Neri y Louis los que tienen la motivación de obtener dinero.

Los niveles de dinero de los personajes han sido renivelados también, de manera que los personajes con la motivación de obtener dinero no necesitarán robar en tantas ocasiones como anteriormente. De esta forma, los planes no se recargarán con acciones repetitivas.

Respecto a las metas, el *Restaurant-Louis* debe quedar destruido, los personajes de Capitán O'Neill y Michael muertos, y Al Neri en la cárcel.

6.7.2 Resultados

Para este problema lo primero que hay que destacar es que en general los planes más interesantes han sido los más aleatorios, al contrario de cómo ocurría anteriormente. Las acciones innecesarias que aparecían en los problemas anteriores ahora se reducen al

mínimo, debido en gran parte a los cambios comentados en la introducción del problema. Los cambios entre los distintos planes generados con este nivel de aleatoriedad son notables. Por otra parte, pese a que los planes generados para los otros niveles de aleatoriedad no tienen el mismo interés, no podemos decir que el resultado de la ejecución sea negativo.

El plan más interesante, obtenido con aleatoriedad alta, es el siguiente:

MICHAEL betrays VITO and quits CORLEONE-FAMILY
SOLLOZZO decide to kill VITO for VENGEANCE
ALNERI goes from HOUSE-ALNERI to HOUSE-CORLEONE to kiss KAY
SOLLOZZO goes from MARKET to HOUSE-LUCABRASI and picks up REVOLVER to kill VITO
SOLLOZZO goes from HOUSE-LUCABRASI to HOUSE-CORLEONE to kill VITO
ALNERI picks up MACHINE-GUN at HOUSE-CORLEONE to rob
VITO command ALNERI to kill MICHAEL for TREASON
VITO lends money to LOUIS
CORLEONE-FAMILY destroys RESTAURANT-LOUIS, LOUIS's business, due to an unpaid debt with VITO
ALNERI kills MICHAEL with MACHINE-GUN at HOUSE-CORLEONE by VITO's command
CAPTAIN-MCCLUSKEY suspects that VITO killed MICHAEL
ALNERI drops MACHINE-GUN at HOUSE-CORLEONE
CAPTAIN-ONEILL suspects that VITO killed MICHAEL
SOLLOZZO kills VITO with REVOLVER at HOUSE-CORLEONE
SOLLOZZO drops REVOLVER at HOUSE-CORLEONE
CAPTAIN-ONEILL suspects that SOLLOZZO killed VITO
CAPTAIN-ONEILL silently chases SOLLOZZO from TRAIN-STATION to HOUSE-CORLEONE
CAPTAIN-ONEILL finds at HOUSE-CORLEONE an evidence on MACHINE-GUN that incriminates ALNERI
ALNERI goes from HOUSE-CORLEONE to HOUSE-BARZINI and picks up RIFLE to rob
CAPTAIN-ONEILL finds at HOUSE-CORLEONE an evidence on REVOLVER that incriminates SOLLOZZO
ALNERI goes from HOUSE-BARZINI to HOUSE-CORLEONE to kiss KAY
CAPTAIN-ONEILL captures ALNERI at HOUSE-CORLEONE and sends him to JAIL
CAPTAIN-ONEILL captures SOLLOZZO at HOUSE-CORLEONE and sends him to JAIL
ALNERI escapes from JAIL and goes to HOUSE-SOLLOZZO
ALNERI goes from HOUSE-SOLLOZZO to HOUSE-CORLEONE to kiss KAY
ALNERI decide to kill CAPTAIN-ONEILL for VENGEANCE
ALNERI kills CAPTAIN-ONEILL with RIFLE at HOUSE-CORLEONE
ALNERI drops RIFLE at HOUSE-CORLEONE
CAPTAIN-MCCLUSKEY suspects that ALNERI killed CAPTAIN-ONEILL
CAPTAIN-MCCLUSKEY silently chases ALNERI from POLICESTATION to HOUSE-CORLEONE
CAPTAIN-MCCLUSKEY finds at HOUSE-CORLEONE an evidence on RIFLE that incriminates ALNERI
CAPTAIN-MCCLUSKEY captures ALNERI at HOUSE-CORLEONE and sends him to JAIL

Aleatoriedad alta						
	1	2	3	4	5	Media
Genera plan	Si	Si	Si	Si	Plan parcial	
Longitud	33	30	22	27	8	28
Diversidad	-	26	10	11	13	15,67
Nº acciones distintas	15	14	13	12	7	13,5
Aleatoriedad media						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	29	28	27	26	31	28,2
Diversidad	-	27	24	20	31	25,5
Nº acciones distintas	14	14	15	14	16	14,6
Aleatoriedad baja						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Sí	Sí	
Longitud	25	25	26	30	25	26,2
Diversidad	-	4	13	26	21	16
Nº acciones distintas	14	14	14	17	14	14,6

Tabla 30. Datos problema 5

6.8 PROBLEMA 6

6.8.1 Definición

Con este problema se pretende que el planificador genere historias complejas, de manera que se compruebe el correcto funcionamiento del sistema al generar narrativas de mayor extensión. Además, se probará el funcionamiento de acciones y predicados que aún no han aparecido en las narrativas generadas. Este es el caso de la inclusión de los guardaespaldas, y los secuestros y rescates. Los predicados añadidos a este problema serán los siguientes:

```
(kidnapped-by michael barzini)
(non-movable michael)
(bodyguard lucabrasi vito)
(has-bodyguard vito)
(has-object alneri gun)
(has-motive barzini vito vengeance)
(likes sollozzo marie)
(debt louis lucabrasi)
(debt michael barzini)
```

De esta forma, presumiblemente Michael será liberado de su secuestro. Además, la inclusión de deudas entre personajes y del predicado *likes* hará que se produzcan confrontaciones. Asimismo, el motivo de asesinato de Barzini para con Vito puede

ocasionar una trama interesante, sobre todo teniendo en cuenta que este último tiene guardaespaldas.

Los predicados del estado meta son los siguientes:

```
(boss-organization michael corleone-family)
(at nino jail)
(kissed sollozzo marie)
(dead lucabراسي)
(at michael policestation)
```

6.8.2 Resultados

Lo primero que observamos en los planes obtenidos es su longitud. En estos planes intervienen gran cantidad de acciones y objetos, y para llegar a las condiciones finales se debe pasar por muchos nodos.

Se han obtenido planes interesantes para todos los niveles de aleatoriedad. Para los niveles alto y medio no en todas las ejecuciones se ha obtenido un plan válido. Sin embargo, incluso en algunas ocasiones un plan no válido puede tener un gran interés narrativo, como ocurre con el siguiente:

```
SOLLOZZO goes from MARKET to JEWELRY-STORE-NINO to kiss MARIE
BARZINI command SOLLOZZO to kill VITO for VENGEANCE
SOLLOZZO goes from JEWELRY-STORE-NINO to HOUSE-CORLEONE and picks up
MACHINE-GUN to kill VITO by BARZINI's command
VITO pays BARZINI for the release of his son MICHAEL
MICHAEL decide to kill BARZINI for VENGEANCE
SOLLOZZO goes from HOUSE-CORLEONE to JEWELRY-STORE-NINO to kiss MARIE
SOLLOZZO kisses MARIE at JEWELRY-STORE-NINO
MICHAEL goes from POLICESTATION to HOUSE-BARZINI and picks up RIFLE to kill
BARZINI
MICHAEL drops RIFLE at HOUSE-BARZINI
SOLLOZZO goes from JEWELRY-STORE-NINO to HOUSE-LUCABRASI to kill
LUCABRASI , VITO's bodyguard, by BARZINI's command
SOLLOZZO kills LUCABRASI, VITO's bodyguard, with HOUSE-LUCABRASI at BARZINI
SOLLOZZO goes from HOUSE-LUCABRASI to HOUSE-CORLEONE to kill VITO by
BARZINI's command
SOLLOZZO kills VITO with MACHINE-GUN at HOUSE-CORLEONE by BARZINI's
command
NINO discovers that SOLLOZZO kissed his lover MARIE and wants to get revenge on them
NINO decide to kill SOLLOZZO for JEALOUSY
CAPTAIN-MCCLUSKEY suspects that BARZINI killed VITO
NINO decide to kill MARIE for JEALOUSY
CAPTAIN-MCCLUSKEY silently chases BARZINI from POLICESTATION to HOUSE-
BARZINI
MICHAEL becomes new boss of CORLEONE-FAMILY
MICHAEL picks up RIFLE at HOUSE-BARZINI to kill BARZINI
MICHAEL kills BARZINI with RIFLE at HOUSE-BARZINI
MICHAEL drops RIFLE at HOUSE-BARZINI
NINO goes from JEWELRY-STORE-NINO to HOUSE-BARZINI and picks up RIFLE to kill
SOLLOZZO
NINO goes from HOUSE-BARZINI to HOUSE-CORLEONE to kill SOLLOZZO
NINO kills SOLLOZZO with RIFLE at HOUSE-CORLEONE
NINO drops RIFLE at HOUSE-CORLEONE
CAPTAIN-MCCLUSKEY suspects that NINO killed SOLLOZZO
CAPTAIN-MCCLUSKEY silently chases NINO from HOUSE-BARZINI to HOUSE-
CORLEONE
CAPTAIN-MCCLUSKEY finds at HOUSE-CORLEONE an evidence on RIFLE that
```

incriminates NINO
 NINO goes from HOUSE-CORLEONE to HOUSE-LUCABRASI and picks up REVOLVER to kill MARIE
 CAPTAIN-MCCLUSKEY silently chases NINO from HOUSE-CORLEONE to HOUSE-LUCABRASI
 CAPTAIN-MCCLUSKEY captures NINO at HOUSE-LUCABRASI and sends him to JAIL
 NINO escapes from JAIL and goes to TRAIN-STATION
 NINO decide to kill CAPTAIN-MCCLUSKEY for VENGEANCE
 NINO goes from TRAIN-STATION to POLICESTATION to kill CAPTAIN-MCCLUSKEY
 NINO drops REVOLVER at POLICESTATION
 NINO goes from POLICESTATION to HOUSE-SOLLOZZO and picks up SHOTGUN to kill MARIE
 NINO goes from HOUSE-SOLLOZZO to HOUSE-CORLEONE to kill CAPTAIN-MCCLUSKEY
 NINO kills CAPTAIN-MCCLUSKEY with SHOTGUN at HOUSE-CORLEONE
 NINO drops SHOTGUN at HOUSE-CORLEONE
 CAPTAIN-ONEILL suspects that NINO killed CAPTAIN-MCCLUSKEY
 CAPTAIN-ONEILL silently chases NINO from TRAIN-STATION to HOUSE-CORLEONE
 CAPTAIN-ONEILL finds at HOUSE-CORLEONE an evidence on SHOTGUN that incriminates NINO
 CAPTAIN-ONEILL captures NINO at HOUSE-CORLEONE and sends him to JAIL

Los planes completos obtenidos también tienen interés narrativo, destacando aquellos que incluyen el menor número de acciones irrelevantes, tales como desplazamientos innecesarios. De entre los planes obtenidos, destaca el siguiente:

SOLLOZZO goes from MARKET to JEWELRY-STORE-NINO to kiss MARIE
 BARZINI command SOLLOZZO to kill VITO for VENGEANCE
 SOLLOZZO goes from JEWELRY-STORE-NINO to HOUSE-CORLEONE and picks up MACHINE-GUN to kill VITO by BARZINI's command
 VITO pays BARZINI for the release of his son MICHAEL
 SOLLOZZO goes from HOUSE-CORLEONE to JEWELRY-STORE-NINO to kiss MARIE
 MICHAEL decide to kill BARZINI for VENGEANCE
 SOLLOZZO kisses MARIE at JEWELRY-STORE-NINO
 NINO discovers that SOLLOZZO kissed his lover MARIE and wants to get revenge on them
 SOLLOZZO goes from JEWELRY-STORE-NINO to HOUSE-LUCABRASI to kill LUCABRASI , VITO's bodyguard, by BARZINI's command
 NINO decide to kill MARIE for JEALOUSY
 SOLLOZZO kills LUCABRASI, VITO's bodyguard, with HOUSE-LUCABRASI at BARZINI
 SOLLOZZO goes from HOUSE-LUCABRASI to HOUSE-CORLEONE to kill VITO by BARZINI's command
 SOLLOZZO kills VITO with MACHINE-GUN at HOUSE-CORLEONE by BARZINI's command
 MICHAEL becomes new boss of CORLEONE-FAMILY
 CAPTAIN-MCCLUSKEY suspects that BARZINI killed LUCABRASI
 CAPTAIN-MCCLUSKEY silently chases BARZINI from POLICESTATION to HOUSE-BARZINI
 NINO goes from JEWELRY-STORE-NINO to HOUSE-BARZINI and picks up RIFLE to kill MARIE
 NINO goes from HOUSE-BARZINI to JEWELRY-STORE-NINO to kill MARIE
 NINO kills MARIE with RIFLE at JEWELRY-STORE-NINO
 NINO drops RIFLE at JEWELRY-STORE-NINO
 CAPTAIN-MCCLUSKEY suspects that NINO killed MARIE
 CAPTAIN-MCCLUSKEY silently chases NINO from HOUSE-BARZINI to JEWELRY-STORE-NINO
 CAPTAIN-MCCLUSKEY finds at JEWELRY-STORE-NINO an evidence on RIFLE that incriminates NINO
 CAPTAIN-MCCLUSKEY captures NINO at JEWELRY-STORE-NINO and sends him to JAIL

Aleatoriedad alta						
	1	2	3	4	5	Media
Genera plan	Plan parcial	Sí	Sí	Sí	Sí	
Longitud	15	37	47	54	30	42
Diversidad	-	44	55	72	74	61,25
Nº acciones distintas	10	16	14	17	15	15,5
Aleatoriedad media						
	1	2	3	4	5	Media
Genera plan	Sí	Sí	Sí	Plan parcial	Sí	
Longitud	53	24	34	17	53	41
Diversidad	-	48	36	33	48	44
Nº acciones distintas	18	15	14	10	18	16.25
Aleatoriedad baja						
	1	2	3	4	5	Media
Genera plan	Plan parcial	Sí	Sí	Sí	Sí	
Longitud	44	31	24	22	29	26,5
Diversidad	-	51	22	24	29	31,5
Nº acciones distintas	16	12	14	13	14	13,25

Tabla 31. Datos problema 6

6.9 PROBLEMA 7

6.9.1 Definición

La última de las pruebas realizadas tiene como objetivo comprobar cómo influye en las historias la inclusión de más personajes. Para ello, y tomando como base los problemas anteriores, se ha construido un nuevo problema en el que se incluyen cinco nuevos personajes. Asimismo, para evitar que la ejecución tarde demasiado tiempo se ha reducido el número de lugares a los que los personajes pueden viajar.

Los personajes introducidos y los cambios asociados a los mismos son los siguientes:

- Vincenzo
- Johnny
 - (in-organization johnny barzini-family)
- Anthony
- Falcone
- Sonny
 - (father-son vito sonny)
 - (in-organization sonny corleone-family)

Se ha incluido además la siguiente información nueva en el estado inicial:

- (debt Vincenzo vito)
- (likes falcone kay)
- (likes sollozzo amy)
- (likes anthony amy)
- (likes sonny amy)
- (likes alneri marie)
- (commanded-to-kill sonny anthony vito)
- (motivation-money vincenzo)
- (motivation-money johnny)
- (motivation-money louis)
- (bodyguard johnny barzini)
- (has-bodyguard barzini)
- (bodyguard lucabراسي vito)
- (has-bodyguard vito)

De esta forma, añadiendo gran cantidad de predicados asociados a todos los personajes en el estado inicial se asegura que en los planes intervendrá un gran número de personajes.

Se ha incluido también una gran cantidad de metas:

- (dead louis)
- (dead vito)
- (dead anthony)
- (dead captain-oneill)
- (dead kay)
- (at michael jail)
- (has-money-level vincenzo highmoney)
- (at nino jail)
- (destroyed restaurant-louis)

6.9.2 Resultados

Cabe destacar acerca de los resultados de este problema es su bajo índice de planes obtenidos que alcanzan todas las metas. Este hecho se debe a la inclusión de más personajes y, sobre todo, a la gran cantidad de metas que se deben alcanzar para que el plan sea válido. Sin embargo, los planes parciales obtenidos con este problema tienen tanto interés como los planes válidos. A continuación se muestra uno de los planes parciales.

```
FALCONE lends money to VINCENZO
ANTHONY goes from RESTAURANT-LOUIS to CASINO to kiss AMY
FALCONE goes from CASINO to HOUSE-CORLEONE to kiss KAY
SONNY goes from HOUSE-CORLEONE to CASINO to kiss AMY
SONNY goes from CASINO to TRAIN-STATION and picks up REVOLVER to kill
ANTHONY by VITO's command
FALCONE kisses KAY at HOUSE-CORLEONE
LUCABRASI, VITO's bodyguard, betrays him and quits CORLEONE-FAMILY
```

VINCENZO goes from TRAIN-STATION to HOUSE-BARZINI and picks up SHOTGUN to rob

VINCENZO goes from HOUSE-BARZINI to BANK to commit a bank robbery

MICHAEL betrays VITO and quits CORLEONE-FAMILY

SONNY goes from TRAIN-STATION to CASINO to kill ANTHONY by VITO's command

VITO command SONNY to kill LUCABRASI for TREASON

SONNY tries to kill ANTHONY at CASINO by VITO's command, but fails. ANTHONY gets hurt

ANTHONY decide to kill VITO for VENGEANCE

MICHAEL discovers that FALCONE kissed his lover KAY and wants to get revenge on them

ANTHONY goes from CASINO to HOUSE-CORLEONE and picks up MACHINE-GUN to kill VITO

SONNY goes from CASINO to TRAIN-STATION to kill LUCABRASI by VITO's command

BARZINI lends money to LOUIS

BARZINI-FAMILY destroys RESTAURANT-LOUIS, LOUIS's business, due to an unpaid debt with BARZINI

LOUIS goes from RESTAURANT-LOUIS to CASINO and picks up KNIFE to rob

VINCENZO goes from BANK to JEWELRY-STORE-NINO, NINO's business, to commit a robbery

VINCENZO robs at JEWELRY-STORE-NINO, NINO's business

SONNY goes from TRAIN-STATION to HOUSE-CORLEONE to kill ANTHONY by VITO's command

MICHAEL decide to kill FALCONE for JEALOUSY

MICHAEL decide to kill KAY for JEALOUSY

ANTHONY kills VITO with MACHINE-GUN at HOUSE-CORLEONE

CAPTAIN-MCCLUSKEY suspects that ANTHONY killed VITO

SONNY goes from HOUSE-CORLEONE to TRAIN-STATION to kill LUCABRASI by VITO's command

LOUIS goes from CASINO to JEWELRY-STORE-NINO, NINO's business, to commit a robbery

CAPTAIN-MCCLUSKEY silently chases ANTHONY from POLICESTATION to HOUSE-CORLEONE

NINO decide to kill VINCENZO for VENGEANCE

MICHAEL wants to get revenge on ANTHONY for killing his father VITO

MICHAEL decide to kill ANTHONY for VENGEANCE

CAPTAIN-ONEILL suspects that ANTHONY killed VITO

VINCENZO goes from JEWELRY-STORE-NINO to BANK to commit a bank robbery

MICHAEL goes from HOUSE-CORLEONE to JEWELRY-STORE-NINO and picks up RIFLE to kill FALCONE

MICHAEL goes from JEWELRY-STORE-NINO to HOUSE-CORLEONE to kill ANTHONY

MICHAEL kills FALCONE with RIFLE at HOUSE-CORLEONE

MICHAEL drops RIFLE at HOUSE-CORLEONE

MICHAEL becomes new boss of CORLEONE-FAMILY

LOUIS robs at JEWELRY-STORE-NINO, NINO's business

MICHAEL picks up RIFLE at HOUSE-CORLEONE to kill ANTHONY

VINCENZO commits a bank robbery at BANK

MICHAEL kills KAY with RIFLE at HOUSE-CORLEONE

MICHAEL drops RIFLE at HOUSE-CORLEONE

NINO decide to kill LOUIS for VENGEANCE

NINO goes from JEWELRY-STORE-NINO to HOUSE-CORLEONE and picks up RIFLE to kill VINCENZO

NINO goes from HOUSE-CORLEONE to JEWELRY-STORE-NINO to kill LOUIS

VINCENZO goes from BANK to JEWELRY-STORE-NINO, NINO's business, to commit a robbery

CAPTAIN-ONEILL silently chases ANTHONY from TRAIN-STATION to HOUSE-CORLEONE

NINO kills LOUIS with RIFLE at JEWELRY-STORE-NINO

ANTHONY drops MACHINE-GUN at HOUSE-CORLEONE

CAPTAIN-ONEILL silently chases VINCENZO from TRAIN-STATION to JEWELRY-

STORE-NINO
 NINO drops RIFLE at JEWELRY-STORE-NINO
 MICHAEL goes from HOUSE-CORLEONE to JEWELRY-STORE-NINO and picks up RIFLE to kill ANTHONY
 MICHAEL drops RIFLE at JEWELRY-STORE-NINO
 CAPTAIN-ONEILL finds at JEWELRY-STORE-NINO an evidence on RIFLE that incriminates MICHAEL
 MICHAEL goes from JEWELRY-STORE-NINO to HOUSE-CORLEONE and picks up MACHINE-GUN to kill ANTHONY
 MICHAEL kills ANTHONY with MACHINE-GUN at HOUSE-CORLEONE
 CAPTAIN-ONEILL captures MICHAEL at HOUSE-CORLEONE and sends him to JAIL
 SONNY drops REVOLVER at TRAIN-STATION
 MICHAEL escapes from JAIL and goes to BANK
 NINO goes from JEWELRY-STORE-NINO to TRAIN-STATION and picks up REVOLVER to kill VINCENZO
 CAPTAIN-MCCLUSKEY suspects that MICHAEL killed FALCONE
 MICHAEL drops MACHINE-GUN at BANK
 NINO goes from TRAIN-STATION to JEWELRY-STORE-NINO to kill VINCENZO
 MICHAEL command SONNY to kill CAPTAIN-ONEILL for VENGEANCE
 NINO kills VINCENZO with REVOLVER at JEWELRY-STORE-NINO
 NINO drops REVOLVER at JEWELRY-STORE-NINO
 CAPTAIN-MCCLUSKEY silently chases MICHAEL from POLICESTATION to BANK
 CAPTAIN-MCCLUSKEY finds at BANK an evidence on MACHINE-GUN that incriminates MICHAEL
 CAPTAIN-MCCLUSKEY captures MICHAEL at BANK and sends him to JAIL
 CAPTAIN-ONEILL finds at JEWELRY-STORE-NINO an evidence on REVOLVER that incriminates NINO

Los planes que alcanzan las metas también tienen interés narrativo, como se observa en el siguiente:

ANTHONY goes from RESTAURANT-LOUIS to CASINO to kiss AMY
 FALCONE goes from CASINO to HOUSE-CORLEONE to kiss KAY
 SONNY goes from HOUSE-CORLEONE to CASINO to kiss AMY
 SONNY goes from CASINO to TRAIN-STATION and picks up REVOLVER to kill ANTHONY by VITO's command
 LOUIS goes from RESTAURANT-LOUIS to JEWELRY-STORE-NINO and picks up RIFLE to rob
 SONNY drops REVOLVER at TRAIN-STATION
 SONNY goes from TRAIN-STATION to HOUSE-CORLEONE and picks up MACHINE-GUN to kill ANTHONY by VITO's command
 FALCONE kisses KAY at HOUSE-CORLEONE
 SONNY goes from HOUSE-CORLEONE to CASINO to kill ANTHONY by VITO's command
 JOHNNY goes from HOUSE-BARZINI to TRAIN-STATION and picks up REVOLVER to rob
 LOUIS drops RIFLE at JEWELRY-STORE-NINO
 JOHNNY goes from TRAIN-STATION to BANK to commit a bank robbery
 JOHNNY drops REVOLVER at BANK
 BARZINI lends money to LOUIS
 LOUIS goes from JEWELRY-STORE-NINO to BANK and picks up REVOLVER to rob
 LOUIS drops REVOLVER at BANK
 LOUIS goes from BANK to HOUSE-BARZINI and picks up SHOTGUN to rob
 LOUIS goes from HOUSE-BARZINI to BANK to commit a bank robbery
 LUCABRASI, VITO's bodyguard, betrays him and quits CORLEONE-FAMILY
 MICHAEL betrays VITO and quits CORLEONE-FAMILY
 SONNY kills ANTHONY with MACHINE-GUN at CASINO by VITO's command
 VINCENZO goes from TRAIN-STATION to CASINO and picks up KNIFE to rob
 BARZINI-FAMILY destroys RESTAURANT-LOUIS, LOUIS's business, due to an unpaid debt with BARZINI

LOUIS goes from BANK to JEWELRY-STORE-NINO, NINO's business, to commit a robbery
 VINCENZO goes from CASINO to JEWELRY-STORE-NINO, NINO's business, to commit a robbery
 VITO command SONNY to kill MICHAEL for TREASON
 LOUIS goes from JEWELRY-STORE-NINO to BANK to commit a bank robbery
 KAY lends money to VINCENZO
 LOUIS decide to kill BARZINI for VENGEANCE
 SONNY goes from CASINO to HOUSE-CORLEONE to kill MICHAEL by VITO's command
 LOUIS commits a bank robbery at BANK
 LOUIS kills JOHNNY, BARZINI's bodyguard, with SHOTGUN at BANK
 CAPTAIN-MCCLUSKEY suspects that VITO killed ANTHONY
 CAPTAIN-MCCLUSKEY suspects that LOUIS killed JOHNNY
 CAPTAIN-MCCLUSKEY silently chases VITO from POLICESTATION to HOUSE-CORLEONE
 LOUIS goes from BANK to JEWELRY-STORE-NINO, NINO's business, to commit a robbery
 CAPTAIN-MCCLUSKEY silently chases LOUIS from HOUSE-CORLEONE to JEWELRY-STORE-NINO
 SONNY tries to kill MICHAEL at HOUSE-CORLEONE by VITO's command, but fails. MICHAEL gets hurt
 VINCENZO robs at JEWELRY-STORE-NINO, NINO's business
 LOUIS robs at JEWELRY-STORE-NINO, NINO's business
 VINCENZO goes from JEWELRY-STORE-NINO to BANK to commit a bank robbery
 MICHAEL decide to kill VITO for VENGEANCE
 NINO decide to kill LOUIS for VENGEANCE
 MICHAEL discovers that FALCONE kissed his lover KAY and wants to get revenge on them
 MICHAEL decide to kill KAY for JEALOUSY
 LOUIS drops SHOTGUN at JEWELRY-STORE-NINO
 CAPTAIN-ONEILL silently chases LOUIS from TRAIN-STATION to JEWELRY-STORE-NINO
 CAPTAIN-ONEILL finds at JEWELRY-STORE-NINO an evidence on SHOTGUN that incriminates LOUIS
 VINCENZO commits a bank robbery at BANK
 LOUIS picks up RIFLE at JEWELRY-STORE-NINO to kill BARZINI
 CAPTAIN-MCCLUSKEY silently chases VINCENZO from HOUSE-CORLEONE to BANK
 NINO goes from JEWELRY-STORE-NINO to BANK and picks up REVOLVER to kill LOUIS
 NINO drops REVOLVER at BANK
 LOUIS goes from JEWELRY-STORE-NINO to BANK to commit a bank robbery
 MICHAEL goes from HOUSE-CORLEONE to BANK and picks up REVOLVER to kill KAY
 LOUIS drops RIFLE at BANK
 MICHAEL goes from BANK to HOUSE-CORLEONE to kill VITO
 MICHAEL kills VITO with REVOLVER at HOUSE-CORLEONE
 MICHAEL kills KAY with REVOLVER at HOUSE-CORLEONE
 NINO picks up RIFLE at BANK to kill LOUIS
 MICHAEL drops REVOLVER at HOUSE-CORLEONE
 CAPTAIN-MCCLUSKEY finds at HOUSE-CORLEONE an evidence on REVOLVER that incriminates MICHAEL
 CAPTAIN-ONEILL silently chases LOUIS from TRAIN-STATION to BANK
 NINO kills LOUIS with RIFLE at BANK
 NINO drops RIFLE at BANK
 NINO decide to kill VINCENZO for VENGEANCE
 CAPTAIN-ONEILL finds at BANK an evidence on RIFLE that incriminates NINO
 VINCENZO drops KNIFE at BANK
 CAPTAIN-ONEILL captures NINO at BANK and sends him to JAIL
 CAPTAIN-MCCLUSKEY captures MICHAEL at HOUSE-CORLEONE and sends him to JAIL
 NINO escapes from JAIL and goes to CASINO
 NINO decide to kill CAPTAIN-ONEILL for VENGEANCE

NINO goes from CASINO to BANK and picks up KNIFE to kill CAPTAIN-ONEILL
 NINO kills CAPTAIN-ONEILL with KNIFE at BANK
 NINO drops KNIFE at BANK
 CAPTAIN-MCCLUSKEY finds at BANK an evidence on KNIFE that incriminates NINO
 CAPTAIN-MCCLUSKEY captures NINO at BANK and sends him to JAIL

Como es posible observar, estos planes tienen un gran nivel de detalle y muchas acciones distintas, siendo muy variados. Debido al alto índice de planes parciales, en la tabla de resultados se han marcado en color rojo las medias obtenidas (calculadas excepcionalmente utilizando datos de planes válidos y planes parciales), indicando que las mismas no tienen la misma fiabilidad que las de los anteriores problemas.

Aleatoriedad alta						
	1	2	3	4	5	Media
Genera plan	Plan parcial	Plan parcial	Plan parcial	Plan parcial	Plan parcial	
Longitud	73	55	50	90	53	64.2
Diversidad	-	106	73	110	119	102
Nº acciones distintas	14	18	17	19	15	16.6
Aleatoriedad media						
	1	2	3	4	5	Media
Genera plan	Plan parcial	Plan parcial	Sí	Plan parcial	Sí	
Longitud	73	59	77	47	63	63.8
Diversidad	-	100	100	96	87	95.75
Nº acciones distintas	16	17	19	17	17	17.2
Aleatoriedad baja						
	1	2	3	4	5	Media
Genera plan	Sí	Plan parcial	Sí	Plan parcial	Plan parcial	
Longitud	75	63	60	30	58	57.2
Diversidad	-	76	81	54	62	68.25
Nº acciones distintas	17	17	17	11	17	15.8

Tabla 32. Datos problema 7

6.10 COMENTARIOS ACERCA DE LOS RESULTADOS

Lo primero que hay que comentar de los resultados obtenidos en las pruebas realizadas anteriormente es el alto índice de planes válidos generados respecto al total, del 94% para los primeros seis problemas. Pese a haber utilizado problemas de alta complejidad, el planificador ha sido capaz de resolverlos casi todos, únicamente fallando en 5 ejecuciones. Esto es debido en gran parte a la posibilidad de activar la gran mayoría de los predicados de varias formas distintas y a partir de diferentes acciones, lo que produce planes muy dinámicos y permite que se puedan resolver muchos problemas. También es preciso destacar que en un problema tan complejo y con tantas metas como el problema 7, se haya conseguido resolver en 4 ocasiones y se hayan obtenido planes interesantes independientemente de ello.

Las tablas de datos de ejecución mostradas para cada uno de los problemas pueden aportar información muy útil acerca de cómo la aleatoriedad de la planificación afecta a los planes. Es posible observar cómo la longitud media de los planes obtenidos es mayor cuando más alta es la aleatoriedad. Asimismo, la diversidad entre los distintos planes generados con la misma configuración es considerablemente mayor también cuando se ha seleccionado alta aleatoriedad. Esto se debe a que el planificador no elige en cada momento la mejor acción para alcanzar las metas, sino que con una probabilidad k elige una acción aleatoria, que puede llevar a alargar el plan. Además, dos planes generados con aleatoriedad 0 serán idénticos, por lo que aumentar la aleatoriedad hace que los planes se diferencien más entre sí. Por otra parte, no se observa una relación significativa entre la aleatoriedad y el número de acciones distintas obtenidas. Seguramente esto sea debido a que cuando el planificador escoge una acción de forma aleatoria puede elegir entre todas las disponibles, teniendo en cuenta todas las posibilidades de ejecutar la misma acción con distintos parámetros.

7. GESTIÓN DEL PROYECTO

7.1 PLANIFICACIÓN INICIAL

Al comienzo de este proyecto, el 1 de febrero, se realizó una planificación del tiempo que se tardaría en hacer las diferentes tareas de las que consta. Se ha fijado una jornada de trabajo estimada de 2 horas de lunes a viernes durante los meses de enero y febrero y las 2 últimas semanas de mayo (por coincidencia con exámenes), y de 4 horas durante marzo, abril y mayo. Se ha calculado que el trabajo estaría terminado el 31 de mayo.

El trabajo se ha dividido en varias fases bien diferenciadas. En primer lugar se realiza la fase de estudio del problema. A continuación, se desarrolla el dominio. En tercer lugar se realizará la fase de experimentación, en la que se realizarán las pruebas del dominio desarrollado. Por último, se realizará la documentación de todo el trabajo llevado a cabo.

La fase de estudio tiene varios objetivos principales. Primero, se investigará acerca de trabajos realizados anteriormente en el mismo campo de la planificación automática, y concretamente en su aplicación a la generación de narrativas. También se decidirá qué tecnología se utilizará para desarrollar el dominio y en qué tipo de planificador se realizará la experimentación. Además, durante esta fase se definirá qué tipo de narrativas se generarán y se comenzará a tomar decisiones de diseño generales de cara al posterior modelado.

Durante la fase de implementación y modelado se desarrollará el dominio en PDDL. En primer lugar se definirán los objetos que intervendrán en los planes. A continuación, se determinarán los predicados que se incluirán. Finalmente, se crearán las acciones utilizando los predicados y los objetos anteriormente creados.

La tercera fase del proyecto es la de experimentación. Utilizando el dominio creado y el planificador deseado, y creando una serie de problemas de prueba, se comprobará el correcto funcionamiento del sistema y se evaluarán los resultados obtenidos.

Por último se escribirá la documentación del proyecto. El resultado será un documento de entre 40 y 60 páginas en el que se describirá el trabajo realizado durante el tiempo de desarrollo y se evaluarán los resultados obtenidos. Cabe destacar que esta fase solapará con la experimentación, pues comenzará antes de empezar ésta y será la última en finalizar, debido a que recoge toda la información relativa a las distintas fases del proyecto.

El siguiente diagrama de Gantt muestra la planificación inicial del proyecto, dividido en las cuatro fases comentadas anteriormente.

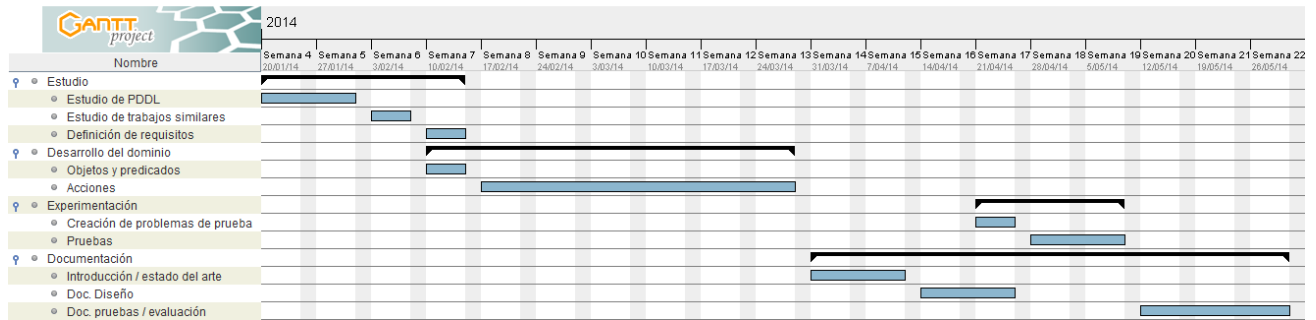


Ilustración 2. Planificación inicial.

7.2 PLANIFICACIÓN FINAL

En el siguiente gráfico se muestra la planificación final de tiempo real invertido durante el desarrollo del proyecto.

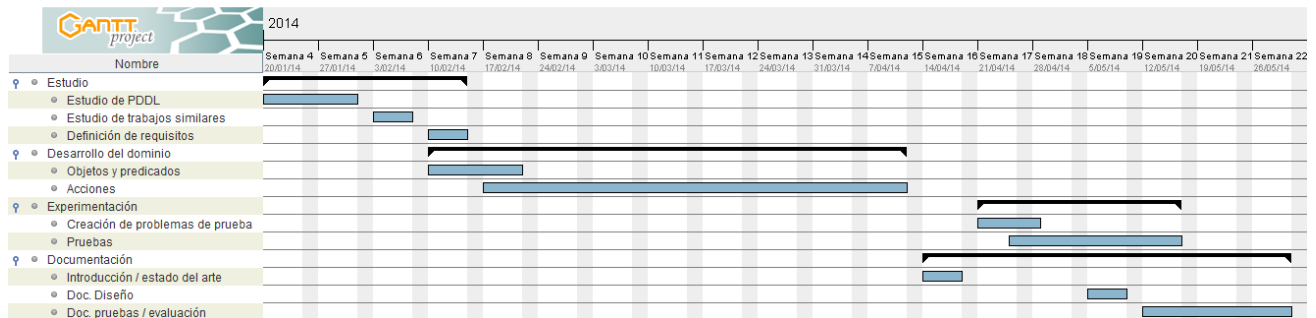


Ilustración 3. Planificación final.

El desarrollo del proyecto ha transcurrido en general como se había planificado, existiendo leves diferencias. En primer lugar, la fase de definición de objetos y predicados se extendió durante más tiempo del previsto, solapando con la fase de creación de acciones, pues al definir acciones se requirió la inclusión de nuevos elementos. A comienzos de la fase de modelado de acciones el conjunto de objetos y predicados terminó de definirse, de manera que los únicos cambios que se necesitó realizar más tarde fueron puntuales. Además, la fase de diseño se alargó más de lo previsto ya que fue necesario añadir acciones que en un principio no se había planeado incluir para que la mayoría de predicados pudieran ser alcanzados por medio de acciones. Además, las pruebas de funcionamiento general que se realizaron periódicamente a medida que se iban incluyendo acciones mostraron que se necesitaban más acciones de las que fueron planeadas en un principio.

Además, las fases documentación de estado del arte y diseño se prolongaron durante algo más tiempo del planeado, como consecuencia en el diseño del tamaño del dominio. También la fase de pruebas tardó más de lo previsto, pues el tiempo que tardaba en ejecutar el sistema con los problemas más complejos era muy alto. Sin embargo, el proyecto se terminó en el plazo esperado, por lo que podemos afirmar que la planificación inicial fue correcta y el ritmo de trabajo aceptable.

7.3 PRESUPUESTO

Los resultados de este trabajo pueden tener un valor económico para empresas o personas involucradas en los sectores de aplicación comentados en la introducción: videojuegos, literatura, cine, etc. Por ello, es importante crear un presupuesto que permita conocer los gastos asociados al desarrollo del trabajo. Para calcular el presupuesto del proyecto se han tenido en cuenta dos tipos diferentes de gastos. En primer lugar se encuentran los gastos de personal, en el que se engloba el dinero destinado al personal que ha trabajado en el proyecto. El gasto de personal por hora utilizado es de 17.18 €.

Fase	Horas de trabajo (h)	Coste (€)
Estudio	35	601.30
Modelado del dominio	105	1803.90
Experimentación	50	859
Documentación	110	1889.80
Total	300	5154

Tabla 33. Gastos de personal

En la siguiente tabla se muestran los gastos referentes a material amortizable, en los que se encuentran productos hardware y productos software.

Tipo	Artículo	Precio (€)	Amortización (meses)	Amortización (€ / mes)	Periodo de uso (meses)	Coste en el proyecto (€)
HW	iMac 27-inch Late 2009	1350.00	60	22.50	4.25	95.63
	Ordenador de sobremesa	1146.00	60	19.10	4.25	81.18
SW	MS Office 2013	539.00	120	4.49	4.25	19.08
	Xcode	0	-	-	-	0
	SAYPHI	0	-	-	-	0
	Emacs	0	-	-	-	0
Total						195.89

Tabla 34. Gastos de material amortizable.

De esta forma, los gastos directos, compuestos por los mencionados anteriormente, son de 5349.89 €. Por otra parte, los gastos indirectos tales como el gasto de luz y agua serán estimados como un 5% de los gastos directos. Asimismo, en caso de venta del trabajo se calcula un margen de beneficios del 5% de los gastos. De esta forma, el presupuesto del total sería el siguiente.

Concepto	Coste
Gastos directos	5349.89
Gastos indirectos	267.50
IVA (21%)	1179.65
Presupuesto total	6797.04

Tabla 35. Presupuesto total.

8. CONCLUSIONES

8.1 COMENTARIOS FINALES

Al comienzo del proyecto el objetivo era crear y modelar un dominio de forma que ejecutado junto con un planificador se obtuvieran historias interesantes, realistas y variadas. Tras analizar los resultados obtenidos en la fase de experimentación, podemos afirmar que el objetivo perseguido se ha conseguido. El conjunto de predicados y acciones del dominio permite que para problemas complejos se obtenga un índice de planes válidos de más del 80%, y para problemas sencillos cercano al 100%.

Desde el punto de vista narrativo, las historias obtenidas son interesantes y podrían ser adaptadas sin problemas al cine, la literatura o los videojuegos. Su estructura, variable entre las distintas historias, se organiza por personajes o por bloques de acciones, como ocurre en los medios comentados anteriormente. De esta forma en las futuras adaptaciones la estructura del resultado final podrá ser directamente la obtenida mediante la planificación. Observando los resultados de la experimentación observamos que los planes obtenidos de problemas sencillos son interesantes también, resultando menos concretos y más generales. Así, un usuario que desee obtener la base de una historia para luego completar y concretar las acciones que van sucediendo creará un problema sencillo, mientras que otro que pretenda obtener todas las acciones de forma concreta, así como la estructura del desarrollo de las mismas, utilizará un problema complejo con muchos elementos.

En cuanto a los problemas que han tenido lugar durante el desarrollo del proyecto, cabe destacar la imposibilidad de utilizar fluents con el planificador elegido. Por ello, para modelar los niveles de dinero que tienen los personajes se utilizaron varios predicados, mediante los cuales se define la jerarquía de niveles de dinero. Este método es menos preciso, pero permite modelar la información deseada de forma correcta utilizando únicamente las posibilidades de PDDL1.2.

8.2 LÍNEAS FUTURAS

Existen muchas posibilidades de mejora para este trabajo dada su naturaleza. En primer lugar, sería posible extender el código del dominio a PDDL3.0, de manera que sea posible obtener historias más complejas. Se podrían incluir fluents para modelar tanto los niveles económicos como las habilidades de los personajes, pudiendo influir estas últimas en el éxito o fracaso de determinadas acciones que llevan a cabo los mismos. También sería posible determinar en los problemas la existencia de objetivos indispensables y objetivos secundarios. Para poder utilizar un dominio que incluya todas estas funcionalidades será necesario encontrar un planificador que las soporte. Esto conllevará un estudio intensivo de varios planificadores.

Sería posible también extender el dominio para que fuera capaz de generar acciones de varios géneros distintos. Varias de las acciones serían comunes a los géneros, mientras que otras serían específicas de cada uno. El usuario escogería en cada problema qué tipo

de historia desea crear, o bien se elegiría aleatoriamente. Así, las posibilidades narrativas serían ostensiblemente mayores.

9. REFERENCIAS

- [1] Malik Ghallab, Dana Nau and Paolo Traverso, “Automated Planning. Theory & Practice”, Morgan Kaufmann, 2004.
- [2] M. Ghallab, A. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld and D. Wilkins, “PDDL – The Planning Domain Definition Language”, 1998, <http://www.informatik.uni-ulm.de/ki/Edu/Vorlesungen/GdKI/WS0203/pddl.pdf>
- [3] Maria Fox and Derek Long, " PDDL2.1 : An Extension to PDDL for Expressing Temporal Planning Domains", December 2003, <http://ipc.informatik.uni-freiburg.de/PddlResources?action=AttachFile&do=get&target=fox-long-jair-2003.pdf>
- [4] Stefan Edelkamp, "PDDL2.2: The Language for the Classical Part of the 4th International Planning Competition", 21th January 2004, <http://ipc.informatik.uni-freiburg.de/PddlResources?action=AttachFile&do=get&target=edelkamp-hoffmann-tr-2004.pdf>
- [5] Alfonso Gerevini and Derek Long, “Preferences and Soft Constraints in PDDL3”, <http://ipc.informatik.uni-freiburg.de/PddlResources?action=AttachFile&do=get&target=gerevini-long-icapsws-2006.pdf>
- [6] Jörg Hoffmann, “The Metric-FF Planning System: Translating ‘Ignoring Delete Lists’ to Numeric State Variables”, Journal of Artificial Intelligence Research, Vol. 20, pp.291-341, 2003.
- [7] Tomás de la Rosa, Ángel García-Olaya and Daniel Borrajo, “A Case-Based Approach to Heuristic Planning”, Applied Intelligence journal, Springer Verlag, Vol. 39, pp.184-201, 2013.
- [8] Patrik Haslum, “Narrative Planning: Compilations to Classical Planning”, Journal of Artificial Intelligence, Vol. 44, pp. 383-395.
- [9] Mark Cavazza, Fred Charles and Steven J. Mead, “Planning Characters’ Behaviour in Interactive Storytelling”, <http://gel.msu.edu/classes/TC848/papers/Cavazza.Planning.Character.Behavior.pdf>
- [10] Julie Porteous, Marc Cavazza and Fred Charles, “Applying Planning to Interactive Storytelling: Narrative Control using State Constraints”, 2010,
- [11] Julie Porteous, Jonathan Teutenberg, Fred Charles and Marc Cavazza, “Controlling Narrative Time in Interactive Storytelling”.

[12] Alicia Fornés Bisquerra, Miguel Lozano Ibáñez. Fernando Barber Miralles, “Formalismos de planificación para la animación comportamental de humanos virtuales”.

ANEXOS I: MANUAL DE USUARIO

En esta sección se proporciona la información necesaria para ejecutar el planificador con el dominio desarrollado y los problemas deseados en una plataforma Mac OS X. También se explicará la forma correcta de crear nuevos problemas, ya que las particularidades del dominio exigen la existencia de ciertos elementos en cada problema que se genere.

En primer lugar, es necesario tener instalado Xcode, el entorno de desarrollo integrado de Apple. Se instalará a continuación MacPorts, que proporciona una manera sencilla de compilar, instalar y actualizar software libre en plataformas con Mac OS X. Ya con MacPorts, se procede a instalar un compilador de CommonLisp. Este trabajo ha sido probado utilizando SBCL. El planificador Sayphi debe ser colocado en el directorio de inicio. Además, es necesario modificar las rutas de los ficheros *.clisprc* o *.sbclrc* y *.init* que se observan a continuación. Será necesario sustituir dichas rutas por el directorio de inicio del computador que se utilice.

.sbclrc

```
;; I load all the common Lisp functions
#+linux (load "/home/pablo/.init.lisp")
#+darwin (load "/Users/pablo/.init.lisp")
#+win32 (load "c:/lisp/.init.lisp")
```

Ilustración 4. Rutas .sbclrc

.clisprc

```
;; I load all the common Lisp functions
#+(or macos darwin) (load "/Users/pablo/.init.lisp")
#+linux (load "/Users/pablo/.init.lisp")
#+win32 (load "c:/lisp/.init.lisp")
```

Ilustración 5. Rutas .clisprc

.init

```
(defvar *my-path*
  #+(or macos darwin) "/Users/pablo/"
  #+linux "/home/pablo/"
  #+win32 "C:/Users/Pablo/")
```

Ilustración 6. Rutas .init

Para ejecutar el planificador con un dominio y un problema concreto es necesario seguir las siguientes instrucciones:

- Abrir una sesión en la Terminal.
- Iniciar una sesión en un compilador de Common Lisp. En el caso de SBCL, se ejecuta la instrucción: *sbcl*. Es posible que la complejidad del dominio y del problema exijan la asignación de mayor cantidad de memoria. En este caso se

ejecutará la instrucción: *sbcl --dynamic-space-size X*, donde *X* es el número de MB de memoria que se desea asignar a la ejecución del programa.

- Cargar los archivos del sistema mediante la instrucción (*load-ma-sayphi*).
- Ejecutar la planificación mediante la instrucción: `(narrative-planner 'rrt carpeta "problem.pddl" :domain-file "domain.pddl" :probability-ehc 0.8 :timeout 300 :number-plans 4 :narrate-p t)`. Los distintos parámetros de la instrucción se explican a continuación:
 - “*carpeta*”: carpeta situada en /home/directorio/planning/sayphi/domains/, en la que se encontrará el dominio. Dentro de la misma se encontrará la carpeta *probsets*, donde estarán los problemas.
 - “*problem1.pddl*”: archivo de problema.
 - “*domain.pddl*”: archivo de dominio.
 - *:probability-ehc 0.8*: probabilidad de que el algoritmo utilizado en la elección de una acción a cada instante no sea aleatorio. Cuanto menor sea el valor, más aleatorios son los planes.
 - *:timeout 300*: tiempo máximo para generar un plan. Si se agota el tiempo, parará la ejecución.
 - *:number-plans 4*: número de planes que se obtienen. Se calcula la diversidad entre los mismos.
 - *:narrate-p t*: si se define el valor “t”, se utilizará la función *narrate* para generar las narrativas. Si no, se mostrarán únicamente las acciones ejecutadas con sus parámetros.

ANEXOS II: CÓDIGO

DOMINIO

```
(define (domain Policiaco)
  (:requirements :strips :typing)
  (:types
    male female - person
    person weapon - locatable
    business bank otherplace policestation - genericplace
    jail genericplace kidnapplace - place
    vengeance jealousy treason - motive
    nomoney lowmoney mediummoney highmoney - moneylevel
    place organization motive locatable moneylevel - object
  )

  (:predicates
    (at ?l - locatable ?place - place)
    (injured ?p1 - person)
    (dead ?p - person)
    (non-movable ?p - person)
    (likes ?p1 - person ?p2 - person)
    (lovers ?p1 - person ?p2 - person)
    (kissed ?p1 - person ?p2 - person)
    (father-son ?p1 - person ?p2 - person)

    (has-weapon ?p1 - person)
    (has-object ?p1 - person ?weapon - weapon)
    (clue-in-object ?person - person ?weapon - weapon)

    (is-police ?p1 - person)
    (suspecting-about ?p1 - person ?p2 - person)

    (has-motive ?p1 - person ?p2 - person ?motive - motive)
    (commanded-to-kill ?killer - person ?victim - person
?commandedkiller - person)
    (intends-to-kill ?killer - person ?victim - person)
    (intends-to-capture ?p1 - person ?p2 - person)
    (intends-to-kidnap ?kidnapper - person ?victim - person)
    (kidnapped-by ?p1 - person ?p2 - person)
    (killed ?killer - person ?victim - person)
    (robbed ?p1 - person ?p2 - person)

    (boss-organization ?p1 - person ?or - organization)
    (is-boss ?p1 - person)
    (successor-organization ?p1 - person ?or - organization)
    (in-organization ?p1 - person ?or - organization)
    (bodyguard ?p1 - person ?p2 - person)
    (has-bodyguard ?p1 - person)

    (has-business ?p1 - person ?b - business)
    (destroyed ?b - business)
    (debt ?p1 - person ?p2 - person)
    (has-money-level ?person - person ?level - moneylevel)
    (next-money-level ?l1 - moneylevel ?l2 - moneylevel)
```

```

(motivation-money ?person - person)

)

;-----MOVEMENT ACTIONS-----
(:action move-to-kill
  :parameters (?killer ?victim - person ?place1 ?place2 -
genericplace)
  :precondition (and
    (not(dead ?victim))
    (not(non-movable ?killer))
    (has-weapon ?killer)
    (intends-to-kill ?killer ?victim)
    (not(has-bodyguard ?victim))
    (at ?killer ?place1)
    (at ?victim ?place2)
    (not(= ?place1 ?place2)))
  :effect (and
    (not(at ?killer ?place1))
    (at ?killer ?place2))
)

(:action move-to-kill-by-command
  :parameters (?killer ?victim ?boss - person ?place1 ?place2
- genericplace)
  :precondition (and
    (not(dead ?victim))
    (not(non-movable ?killer))
    (has-weapon ?killer)
    (commanded-to-kill ?killer ?victim ?boss)
    (not(has-bodyguard ?victim))
    (at ?killer ?place1)
    (not(= ?place1 ?place2))
    (at ?victim ?place2))
  :effect (and
    (not(at ?killer ?place1))
    (at ?killer ?place2))
)

(:action move-to-kill-bodyguard
  :parameters (?killer ?victim ?bg - person ?place1 ?place2 -
genericplace)
  :precondition (and
    (not(dead ?victim))
    (not(non-movable ?killer))
    (not(dead ?bg))
    (has-weapon ?killer)
    (intends-to-kill ?killer ?victim)
    (bodyguard ?bg ?victim)
    (not(= ?place1 ?place2))
    (at ?killer ?place1)
    (at ?bg ?place2))
  :effect (and
    (not(at ?killer ?place1))
    (at ?killer ?place2))
)

```

```

(:action move-to-kill-bodyguard-by-command
  :parameters (?killer ?victim ?bg ?boss - person ?place1
?place2 - genericplace)
  :precondition (and
    (not(dead ?victim))
    (not(non-movable ?killer))
    (not(dead ?bg))
    (has-weapon ?killer)
    (commanded-to-kill ?killer ?victim ?boss)
    (bodyguard ?bg ?victim)
    (not(= ?place1 ?place2))
    (at ?killer ?place1)
    (at ?bg ?place2))
  :effect (and
    (not(at ?killer ?place1))
    (at ?killer ?place2)))

(:action move-to-kidnap
  :parameters (?kidnapper ?victim - person ?place1 ?place2 -
genericplace)
  :precondition (and
    (not(dead ?victim))
    (not(non-movable ?kidnapper))
    (not(= ?place1 ?place2))
    (intends-to-kidnap ?kidnapper ?victim)
    (at ?kidnapper ?place1)
    (at ?victim ?place2))
  :effect (and
    (not(at ?kidnapper ?place1))
    (at ?kidnapper ?place2))
)

(:action move-to-rob-bank
  :parameters (?robber - person ?place1 - genericplace ?bank
- bank)
  :precondition (and
    (at ?robber ?place1)
    (not(non-movable ?robber))
    (has-weapon ?robber)
    (not(= ?place1 ?bank))
    (motivation-money ?robber)
  )
  :effect (and
    (not(at ?robber ?place1))
    (at ?robber ?bank)))

(:action move-to-rob-business
  :parameters (?robber ?victim - person ?place1 - genericplace
?business - business)
  :precondition (and
    (at ?robber ?place1)
    (not(non-movable ?robber))
    (not(dead ?victim))
    (has-weapon ?robber)
    (not(destroyed ?business))
    (not(= ?place1 ?business))
    (has-business ?victim ?business))

```

```

        (motivation-money ?robber)
    )
    :effect (and
        (not(at ?robber ?place1))
        (at ?robber ?business)))

    (:action move-to-kiss
    :parameters (?p1 - male ?p2 - female ?place1 ?place2 -
genericplace)
    :precondition (and
        (not(non-movable ?p1))
        (not(non-movable ?p2))
        (likes ?p1 ?p2)
        (not(= ?place1 ?place2))
        (at ?p1 ?place1)
        (at ?p2 ?place2))
    :effect (and
        (not(at ?p1 ?place1))
        (at ?p1 ?place2))
    )
;-----OBJECT ACTIONS-----
    (:action pick-up-object-to-kill
    :parameters (?person ?person2 - person ?place -
genericplace ?weapon - weapon)
    :precondition (and
        (not(non-movable ?person))
        (not(dead ?person2))
        (at ?person ?place)
        (at ?weapon ?place)
        (intends-to-kill ?person ?person2)
        (not(has-weapon ?person))
    )
    :effect (and
        (not(at ?weapon ?place))
        (has-object ?person ?weapon)
        (has-weapon ?person)))

    (:action pick-up-object-to-kill-by-command
    :parameters (?person ?person2 ?boss - person ?place -
genericplace ?weapon - weapon)
    :precondition (and
        (not(non-movable ?person))
        (at ?person ?place)
        (at ?weapon ?place)
        (commanded-to-kill ?person ?person2 ?boss)
        (not(has-weapon ?person))
    )
    :effect (and
        (not(at ?weapon ?place))
        (has-object ?person ?weapon)
        (has-weapon ?person)))

    (:action pick-up-object-to-rob
    :parameters (?person - person ?place - genericplace ?weapon
- weapon)
    :precondition (and
        (not(non-movable ?person))

```

```

        (at ?person ?place)
        (at ?weapon ?place)
        (motivation-money ?person)
        (not(has-weapon ?person))
    )
    :effect (and
        (not(at ?weapon ?place))
        (has-object ?person ?weapon)
        (has-weapon ?person)))

    (:action move-to-pickup-object-kill
     :parameters (?killer ?victim - person ?place1 ?place2 -
genericplace ?weapon - weapon)
     :precondition (and
        (not(dead ?victim))
        (not(non-movable ?killer))
        (not(has-weapon ?killer))
        (intends-to-kill ?killer ?victim)
        (not(= ?place1 ?place2))
        (at ?killer ?place1)
        (at ?weapon ?place2))
     :effect (and
        (not(at ?killer ?place1))
        (at ?killer ?place2)
        (has-object ?killer ?weapon)
        (has-weapon ?killer)
        (not(at ?weapon ?place2)))
    )

    (:action move-to-pickup-object-kill-by-command
     :parameters (?killer ?victim ?boss - person ?place1 ?place2
- genericplace ?weapon - weapon)
     :precondition (and
        (not(dead ?victim))
        (not(non-movable ?killer))
        (not(has-weapon ?killer))
        (commanded-to-kill ?killer ?victim ?boss)
        (not(= ?place1 ?place2))
        (at ?killer ?place1)
        (at ?weapon ?place2))
     :effect (and
        (not(at ?killer ?place1))
        (at ?killer ?place2)
        (has-object ?killer ?weapon)
        (has-weapon ?killer)
        (not(at ?weapon ?place2)))
    )

    (:action move-to-pickup-object-rob
     :parameters (?robber - person ?place1 ?place2 - genericplace
?weapon - weapon)
     :precondition (and
        (at ?robber ?place1)
        (at ?weapon ?place2)
        (not(non-movable ?robber))
        (not(has-weapon ?robber))
        (motivation-money ?robber)

```

```

        (not(= ?place1 ?place2))
    )
    :effect (and
        (not(at ?robber ?place1))
        (at ?robber ?place2)
        (has-object ?robber ?weapon)
        (has-weapon ?robber)
        (not(at ?weapon ?place2))))

    (:action drop-object
      :parameters (?person - person ?place - genericplace ?weapon
- weapon)
      :precondition (and
        (not(non-movable ?person))
        (at ?person ?place)
        (has-object ?person ?weapon))
      :effect (and
        (not(has-object ?person ?weapon))
        (not(has-weapon ?person))
        (at ?weapon ?place)))

;-----KILLING ACTIONS-----
    (:action decide-murder
      :parameters (?killer ?person - person ?motive - motive)
      :precondition (and
        (has-motive ?killer ?person ?motive)
        (not(non-movable ?person))
        (not(non-movable ?killer))
        (not(is-boss ?killer)))
      :effect (and
        (intends-to-kill ?killer ?person)
        (not(has-motive ?killer ?person ?motive))))

    (:action command-murder
      :parameters (?boss ?person ?killer - person ?motive - motive
?or - organization)
      :precondition (and
        (has-motive ?boss ?person ?motive)
        (boss-organization ?boss ?or)
        (not(non-movable ?person))
        (not(non-movable ?boss))
        (not(non-movable ?killer))
        (in-organization ?killer ?or))
      :effect (and
        (not(has-motive ?boss ?person ?motive))
        (commanded-to-kill ?killer ?person ?boss)))

    (:action kill
      :parameters (?killer ?victim - person ?place - genericplace
?weapon - weapon)
      :precondition (and
        (not(non-movable ?killer))
        (not(non-movable ?victim))
        (intends-to-kill ?killer ?victim)
        (at ?killer ?place)

```



```

        (at ?victim ?place)
        (has-object ?killer ?weapon)
        (not (has-bodyguard ?victim))
    )
    :effect (and
        (dead ?victim)
        (non-movable ?victim)
        (clue-in-object ?killer ?weapon)
        (not (intends-to-kill ?killer ?victim))
        (killed ?killer ?victim))

    (:action kill-by-command
        :parameters (?killer ?victim ?boss - person ?place -
genericplace ?weapon - weapon)
        :precondition (and
            (not (non-movable ?killer))
            (not (non-movable ?victim))
            (not (non-movable ?boss))
            (commanded-to-kill ?killer ?victim ?boss)
            (not (= ?killer ?boss))
            (at ?killer ?place)
            (at ?victim ?place)
            (has-object ?killer ?weapon)
            (not (has-bodyguard ?victim)))
        :effect (and
            (dead ?victim)
            (non-movable ?victim)
            (clue-in-object ?killer ?weapon)
            (not (commanded-to-kill ?killer ?victim ?boss))
            (killed ?boss ?victim)))

    (:action kill-bodyguard
        :parameters (?killer ?victim ?bg - person ?place -
genericplace ?weapon - weapon)
        :precondition (and
            (not (non-movable ?killer))
            (not (non-movable ?victim))
            (not (non-movable ?bg))
            (intends-to-kill ?killer ?victim)
            (at ?killer ?place)
            (at ?bg ?place)
            (has-object ?killer ?weapon)
            (bodyguard ?bg ?victim))
        :effect (and
            (dead ?bg)
            (non-movable ?bg)
            (clue-in-object ?killer ?weapon)
            (not (has-bodyguard ?victim))
            (not (bodyguard ?bg ?victim))
            (killed ?killer ?bg)))

    (:action kill-bodyguard-by-command
        :parameters (?killer ?victim ?bg ?boss - person ?place -
genericplace ?weapon - weapon)
        :precondition (and
            (not (non-movable ?killer))

```

```

        (not(non-movable ?victim))
        (not(non-movable ?bg))
        (commanded-to-kill ?killer ?victim ?boss)
        (not (= ?boss ?killer))
        (at ?killer ?place)
        (at ?bg ?place)
        (has-object ?killer ?weapon)
        (bodyguard ?bg ?victim))
    :effect (and
        (dead ?bg)
        (non-movable ?bg)
        (clue-in-object ?killer ?weapon)
        (not(has-bodyguard ?victim))
        (not(bodyguard ?bg ?victim))
        (killed ?boss ?bg)))

    (:action failed-kill
      :parameters (?killer ?victim - person ?place - genericplace
        ?weapon - weapon ?vengeance - vengeance)
      :precondition (and
        (not(non-movable ?killer))
        (not(non-movable ?victim))
        (not(injured ?victim))
        (intends-to-kill ?killer ?victim)
        (at ?killer ?place)
        (at ?victim ?place)
        (has-object ?killer ?weapon)
        (not(has-bodyguard ?victim)))
      :effect (and
        (injured ?victim)
        (has-motive ?victim ?killer ?vengeance))
    )

    (:action failed-kill-by-command
      :parameters (?killer ?victim ?boss - person ?place -
        genericplace ?weapon - weapon ?vengeance - vengeance)
      :precondition (and
        (not(non-movable ?killer))
        (not(non-movable ?victim))
        (not(injured ?victim))
        (not(non-movable ?boss))
        (not(= ?killer ?boss))
        (commanded-to-kill ?killer ?victim ?boss)
        (at ?killer ?place)
        (at ?victim ?place)
        (has-object ?killer ?weapon))
      :effect (and
        (injured ?victim)
        (has-motive ?victim ?boss ?vengeance))
    )

    (:action vengeance-lover
      :parameters (?victim ?killer ?relative - person ?vengeance
        - vengeance)
      :precondition (and
        (not(dead ?killer))
        (not(non-movable ?relative))

```

```

        (killed ?killer ?victim)
        (lovers ?victim ?relative))
    :effect
        (has-motive ?relative ?killer ?vengeance))

    (:action vengeance-father
     :parameters (?victim ?killer ?relative - person ?vengeance
- vengeance)
     :precondition (and
        (not(dead ?killer))
        (not(non-movable ?relative))
        (killed ?killer ?victim)
        (father-son ?victim ?relative))
     :effect
        (has-motive ?relative ?killer ?vengeance))

;-----EMOTIONAL ACTIONS-----
    (:action kiss
     :parameters (?p1 - male ?p2 - female ?place - genericplace)
     :precondition (and
        (not(non-movable ?p1))
        (not(non-movable ?p2))
        (likes ?p1 ?p2)
        (at ?p1 ?place)
        (at ?p2 ?place)
        (not(lovers ?p1 ?p2)))
     :effect (and
        (kissed ?p1 ?p2)
        (not(likes ?p1 ?p2))
        (lovers ?p1 ?p2)))

    (:action discover-kissing
     :parameters (?p1 ?p3 - male ?p2 - female ?jealousy -
jealousy)
     :precondition (and
        (not(non-movable ?p1))
        (not(non-movable ?p2))
        (not(non-movable ?p3))
        (lovers ?p1 ?p2)
        (kissed ?p3 ?p2)
        (not (= ?p1 ?p3)))
     :effect (and
        (has-motive ?p1 ?p2 ?jealousy)
        (has-motive ?p1 ?p3 ?jealousy)))

```

```

;-----POLICE ACTIONS-----

```

```

(:action find-clue-object
  :parameters (?police ?person - person ?place - genericplace
?weapon - weapon)
  :precondition (and
    (not(non-movable ?person))
    (not(non-movable ?police))
    (at ?police ?place)
    (at ?weapon ?place)
    (is-police ?police)
    (clue-in-object ?person ?weapon))
  :effect (and
    (intends-to-capture ?police ?person)
    (not(clue-in-object ?person ?weapon))
    (not(at ?weapon ?place))))

(:action suspect-police
  :parameters (?police ?killer ?victim - person)
  :precondition (and
    (is-police ?police)
    (not(non-movable ?police))
    (not(non-movable ?killer))
    (killed ?killer ?victim))
  :effect
    (suspecting-about ?police ?killer)
)

(:action follow
  :parameters (?police ?person - person ?place1 ?place2 -
genericplace)
  :precondition (and
    (not(non-movable ?police))
    (not(non-movable ?person))
    (is-police ?police)
    (suspecting-about ?police ?person)
    (at ?police ?place1)
    (at ?person ?place2))
  :effect
    (at ?police ?place2))

(:action capture
  :parameters (?police ?person - person ?place - genericplace
?jail - jail ?vengeance - vengeance)
  :precondition (and
    (not(non-movable ?police))
    (not(non-movable ?person))
    (at ?police ?place)
    (at ?person ?place)
    (is-police ?police)
    (intends-to-capture ?police ?person))
  :effect (and
    (not(at ?person ?place))
    (at ?person ?jail)
    (non-movable ?person)
    (not(suspecting-about ?police ?person))
    (not(intends-to-capture ?police ?person))
    (has-motive ?person ?police ?vengeance)))

```

```

(:action escape-from-jail
  :parameters (?person - person ?jail - jail ?place -
genericplace)
  :precondition (and
    (not(dead ?person))
    (at ?person ?jail))
  :effect (and
    (not(non-movable ?person))
    (not(at ?person ?jail))
    (at ?person ?place)))

```

;-----MAFIA ACTIONS-----

```

(:action new-boss
  :parameters (?boss ?successor - person ?or - organization)
  :precondition (and
    (dead ?boss)
    (not(dead ?successor))
    (boss-organization ?boss ?or)
    (successor-organization ?successor ?or))
  :effect (and
    (not(boss-organization ?boss ?or))
    (boss-organization ?successor ?or)
    (not(successor-organization ?successor ?or)))

```

```

(:action betray
  :parameters (?traitor ?boss - person ?or - organization
?treason - treason)
  :precondition (and
    (not(non-movable ?traitor))
    (not(dead ?boss))
    (in-organization ?traitor ?or)
    (not(has-bodyguard ?boss))
    (boss-organization ?boss ?or))
  :effect (and
    (not(in-organization ?traitor ?or))
    (has-motive ?boss ?traitor ?treason)))

```

```

(:action betray-bodyguard
  :parameters (?traitor ?boss - person ?or - organization
?treason - treason)
  :precondition (and
    (not(non-movable ?traitor))
    (not(dead ?boss))
    (in-organization ?traitor ?or)
    (boss-organization ?boss ?or)
    (bodyguard ?traitor ?boss))
  :effect (and
    (not(bodyguard ?traitor ?boss))
    (not(has-bodyguard ?boss))
    (not(in-organization ?traitor ?or))
    (has-motive ?boss ?traitor ?treason)))

```

```

(:action destroy-business

```

```

        :parameters (?bus - business ?or - organization ?boss ?person
- person ?vengeance - vengeance)
        :precondition (and
            (not(non-movable ?boss))
            (not(non-movable ?person))
            (has-business ?person ?bus)
            (debt ?person ?boss)
            (boss-organization ?boss ?or)
            (not(destroyed ?bus)))
        :effect (and
            (destroyed ?bus)
            (has-motive ?person ?boss ?vengeance)))

```

;-----MONEY ACTIONS-----

```

        (:action loan
        :parameters (?p1 ?p2 - person ?nomoney - nomoney ?lowmoney
- lowmoney ?highmoney - highmoney ?mediummoney - mediummoney)
        :precondition (and
            (has-money-level ?p1 ?nomoney)
            (has-money-level ?p2 ?highmoney))
        :effect (and
            (not(has-money-level ?p1 ?nomoney))
            (has-money-level ?p1 ?lowmoney)
            (not(has-money-level ?p2 ?highmoney))
            (has-money-level ?p2 ?mediummoney)
            (debt ?p1 ?p2)))

```

```

        (:action pay-debt
        :parameters (?p1 ?p2 - person ?p111 ?p112 ?p211 ?p212 -
moneylevel ?nomoney - nomoney)
        :precondition (and
            (not(non-movable ?p1))
            (not(non-movable ?p2))
            (debt ?p1 ?p2)
            (not(has-money-level ?p1 ?nomoney))
            (has-money-level ?p1 ?p111)
            (has-money-level ?p2 ?p211)
            (next-money-level ?p112 ?p111)
            (next-money-level ?p211 ?p212)
        )
        :effect (and
            (not(debt ?p1 ?p2))
            (has-money-level ?p1 ?p112)
            (has-money-level ?p2 ?p212)))

```

```

        (:action decide-kidnap-father-son
        :parameters (?kidnapper ?victim ?relativetovictim - person)
        :precondition (and
            (not(non-movable ?kidnapper))
            (not(non-movable ?victim))
            (not(non-movable ?relativetovictim))
            (debt ?victim ?kidnapper)
            (father-son ?relativetovictim ?victim))
        :effect
            (intends-to-kidnap ?kidnapper ?victim))

```

```

(:action decide-kidnap-lovers
  :parameters (?kidnapper ?victim ?relativetovictim - person)
  :precondition (and
    (not(non-movable ?kidnapper))
    (not(non-movable ?victim))
    (not(non-movable ?relativetovictim))
    (debt ?victim ?kidnapper)
    (lovers ?relativetovictim ?victim))
  :effect
    (intends-to-kidnap ?kidnapper ?victim))

(:action kidnap
  :parameters (?kidnapper ?victim - person ?captureplace -
genericplace ?kidnapplace - kidnappplace ?highmoney - highmoney)
  :precondition (and
    (not(non-movable ?kidnapper))
    (not(non-movable ?victim))
    (at ?kidnapper ?captureplace)
    (at ?victim ?captureplace)
    (not(has-money-level ?kidnapper ?highmoney))
    (intends-to-kidnap ?kidnapper ?victim))
  :effect (and
    (kidnapped-by ?victim ?kidnapper)
    (not(intends-to-kidnap ?kidnapper ?victim))
    (not(at ?victim ?captureplace))
    (non-movable ?victim)
    (at ?victim ?kidnapplace)))

(:action pay-rescue-father
  :parameters (?kidnapper ?victim ?relativetovictim - person
?kidnapplace - kidnappplace ?policestation - policestation ?vengeance -
vengeance ?nomoney - nomoney ?relativell ?relativel2 ?kidnapperl1
?kidnapperl2 - moneylevel)
  :precondition (and
    (debt ?victim ?kidnapper)
    (not(non-movable ?kidnapper))
    (not(dead ?victim))
    (not(non-movable ?relativetovictim))
    (at ?victim ?kidnapplace)
    (kidnapped-by ?victim ?kidnapper)
    (father-son ?relativetovictim ?victim)
    (not(has-money-level ?relativetovictim ?nomoney))
    (next-money-level ?relativel2 ?relativell)
    (next-money-level ?kidnapperl1 ?kidnapperl2)
    (has-money-level ?relativetovictim ?relativell)
    (has-money-level ?kidnapper ?kidnapperl1))
  :effect (and
    (not(debt ?victim ?kidnapper))
    (not(has-money-level ?relativetovictim ?relativell))
    (has-money-level ?relativetovictim ?relativel2)
    (not(has-money-level ?kidnapper ?kidnapperl1))
    (has-money-level ?kidnapper ?kidnapperl2)
    (has-motive ?victim ?kidnapper ?vengeance)
    (not(kidnapped-by ?victim ?kidnapper))
    (not(non-movable ?victim))
    (not(at ?victim ?kidnapplace)))

```

```

(at ?victim ?policestation)))

(:action pay-rescue-lover
  :parameters (?kidnapper ?victim ?relativetovictim - person
?kidnapplace - kidnapplace ?policestation - policestation ?vengeance -
vengeance ?nomoney - nomoney ?relativell ?relativel2 ?kidnapperl1
?kidnapperl2 - moneylevel)
  :precondition (and
    (debt ?victim ?kidnapper)
    (not(non-movable ?kidnapper))
    (not(dead ?victim))
    (not(non-movable ?relativetovictim))
    (at ?victim ?kidnapplace)
    (kidnapped-by ?victim ?kidnapper)
    (lovers ?victim ?relativetovictim)
    (not(has-money-level ?relativetovictim ?nomoney))
    (next-money-level ?relativel2 ?relativell)
    (next-money-level ?kidnapperl1 ?kidnapperl2)
    (has-money-level ?relativetovictim ?relativell)
    (has-money-level ?kidnapper ?kidnapperl1))
  :effect (and
    (not(debt ?victim ?kidnapper))
    (not(has-money-level ?relativetovictim ?relativell))
    (has-money-level ?relativetovictim ?relativel2)
    (not(has-money-level ?kidnapper ?kidnapperl1))
    (has-money-level ?kidnapper ?kidnapperl2)
    (has-motive ?victim ?kidnapper ?vengeance)
    (not(kidnapped-by ?victim ?kidnapper))
    (not(non-movable ?victim))
    (not(at ?victim ?kidnapplace))
    (at ?victim ?policestation)))

(:action decide-rob-lovers
  :parameters (?victim ?kidnapper ?relativetovictim - person
?nomoney - nomoney)
  :precondition (and
    (not(dead ?victim))
    (not(non-movable ?kidnapper))
    (not(non-movable ?relativetovictim))
    (kidnapped-by ?victim ?kidnapper)
    (lovers ?victim ?relativetovictim)
    (has-money-level ?relativetovictim ?nomoney))
  :effect
    (motivation-money ?relativetovictim))

(:action decide-rob-father-son
  :parameters (?victim ?kidnapper ?relativetovictim - person
?nomoney - nomoney)
  :precondition (and
    (not(dead ?victim))
    (not(non-movable ?kidnapper))
    (not(non-movable ?relativetovictim))
    (kidnapped-by ?victim ?kidnapper)
    (father-son ?relativetovictim ?victim)
    (has-money-level ?relativetovictim))

```



```

        :effect
            (motivation-money ?relativetovictim))

    (:action rob-business
      :parameters (?robber ?victim - person ?business - business
?vengeance - vengeance ?robberl1 ?robberl2 ?victiml1 ?victiml2 -
moneylevel)
      :precondition (and
        (at ?robber ?business)
        (not(non-movable ?robber))
        (not(dead ?victim))
        (has-weapon ?robber)
        (has-business ?victim ?business)
        (next-money-level ?robberl1 ?robberl2)
        (next-money-level ?victiml2 ?victiml1)
        (has-money-level ?robber ?robberl1)
        (has-money-level ?victim ?victiml1)
        (not(robbed ?robber ?victim)))
      :effect (and
        (has-motive ?victim ?robber ?vengeance)
        (not(has-money-level ?robber ?robberl1))
        (not(has-money-level ?victim ?victiml1))
        (has-money-level ?robber ?robberl2)
        (has-money-level ?victim ?victiml2)
        (robbed ?robber ?victim)))

    (:action rob-bank
      :parameters (?person ?police - person ?bank - bank ?l1 ?l2
- moneylevel)
      :precondition (and
        (at ?person ?bank)
        (is-police ?police)
        (not(non-movable ?person))
        (has-weapon ?person)
        (motivation-money ?person)
        (next-money-level ?l1 ?l2)
        (has-money-level ?person ?l1))
      :effect (and
        (not(has-money-level ?person ?l1))
        (has-money-level ?person ?l2)
        (suspecting-about ?police ?person))
    )

    (:action no-motivation-money
      :parameters (?p1 - person ?highmoney - highmoney)
      :precondition (and
        (motivation-money ?p1)
        (has-money-level ?p1 ?highmoney))
      :effect (not(motivation-money ?p1)))
  )

```

PROBLEM1.PDDL

```
(define (problem policiaco1) (:domain Policiaco)
(:objects

vito michael lucabراسي alneri sollozzo captain-mccluskey captain-oneill
louis nino barzini - male
restaurant-louis jewelry-store-nino - business
kay amy marie - female
gun machine-gun knife shotgun revolver rifle - weapon
corleone-family barzini-family - organization
jealousy - jealousy
treason - treason
vengeance - vengeance
house-corleone house-alneri house-lucabراسي house-sollozzo house-
barzini house-amy market train-station casino - genericplace
kidnapplace - kidnapplace
jail - jail
policestation - policestation
bank - bank
nomoney - nomoney
lowmoney - lowmoney
mediummoney - mediummoney
highmoney - highmoney
)

(:init

;Fijas-----

(next-money-level nomoney lowmoney)
(next-money-level lowmoney mediummoney)
(next-money-level mediummoney highmoney)

(is-police captain-mccluskey)
(is-police captain-oneill)

(has-business louis restaurant-louis)
(has-business nino jewelry-store-nino)

(boss-organization barzini barzini-family)
(boss-organization vito corleone-family)
(successor-organization michael corleone-family)
(is-boss vito)
(is-boss barzini)
(in-organization sollozzo barzini-family)
(in-organization michael corleone-family)
(in-organization alneri corleone-family)
(in-organization lucabراسي corleone-family)

(lovers michael kay)
(lovers nino marie)
(father-son vito michael)

;Obligatorias y modificables-----

(at vito house-corleone)
```

```

(at michael house-corleone)
(at lucabراس house-lucabراس)
(at alneri house-alneri)
(at sollozzo market)
(at barzini house-barzini)
(at captain-mccluskey policestation)
(at captain-oneill train-station)
(at louis restaurant-louis)
(at marie jewelry-store-nino)
(at nino jewelry-store-nino)
(at amy house-amy)

(at gun restaurant-louis)
(at machine-gun house-corleone)
(at knife market)
(at shotgun house-sollozzo)
(at revolver house-lucabراس)
(at rifle house-barzini)

(has-money-level vito mediummoney)
(has-money-level michael highmoney)
(has-money-level lucabراس lowmoney)
(has-money-level alneri nomoney)
(has-money-level sollozzo nomoney)
(has-money-level barzini mediummoney)
(has-money-level kay nomoney)
(has-money-level captain-mccluskey mediummoney)
(has-money-level captain-oneill nomoney)
(has-money-level louis highmoney)
(has-money-level marie lowmoney)
(has-money-level nino highmoney)
(has-money-level amy mediummoney)

;Añadibles-----

(motivation-money lucabراس)
(motivation-money sollozzo)

)

(:goal
  (dead lucabراس)
)
)

```

PROBLEM2.PDDL

```
(define (problem policiaco1) (:domain Policiaco)
(:objects

vito michael lucabراس alneri sollozzo captain-mccluskey captain-oneill
louis nino barzini - male
restaurant-louis jewelry-store-nino - business
kay amy marie - female
gun machine-gun knife shotgun revolver rifle - weapon
corleone-family barzini-family - organization
jealousy - jealousy
treason - treason
vengeance - vengeance
house-corleone house-alneri house-lucabراس house-sollozzo house-
barzini house-amy market train-station casino - genericplace
kidnapplace - kidnapplace
jail - jail
policestation - policestation
bank - bank
nomoney - nomoney
lowmoney - lowmoney
mediummoney - mediummoney
highmoney - highmoney

)

(:init

;Fijas-----

(next-money-level nomoney lowmoney)
(next-money-level lowmoney mediummoney)
(next-money-level mediummoney highmoney)

(is-police captain-mccluskey)
(is-police captain-oneill)

(has-business louis restaurant-louis)
(has-business nino jewelry-store-nino)

(boss-organization barzini barzini-family)
(boss-organization vito corleone-family)
(successor-organization michael corleone-family)
(is-boss vito)
(is-boss barzini)
(in-organization sollozzo barzini-family)
(in-organization michael corleone-family)
(in-organization alneri corleone-family)
(in-organization lucabراس corleone-family)

(lovers michael kay)
(lovers nino marie)
(father-son vito michael)

;Obligatorias y modificables-----
```

```

(at vito house-corleone)
(at michael house-corleone)
(at lucabراس house-lucabراس)
(at alneri house-alneri)
(at sollozzo market)
(at barzini house-barzini)
(at captain-mccluskey policestation)
(at captain-oneill train-station)
(at louis restaurant-louis)
(at marie jewelry-store-nino)
(at nino jewelry-store-nino)
(at amy house-amy)

(at gun restaurant-louis)
(at machine-gun house-corleone)
(at knife market)
(at shotgun house-sollozzo)
(at revolver house-lucabراس)
(at rifle house-barzini)

(has-money-level vito mediummoney)
(has-money-level michael highmoney)
(has-money-level lucabراس lowmoney)
(has-money-level alneri nomoney)
(has-money-level sollozzo nomoney)
(has-money-level barzini mediummoney)
(has-money-level kay nomoney)
(has-money-level captain-mccluskey mediummoney)
(has-money-level captain-oneill nomoney)
(has-money-level louis highmoney)
(has-money-level marie lowmoney)
(has-money-level nino highmoney)
(has-money-level amy mediummoney)

;Añadibles-----

(motivation-money lucabراس)
(motivation-money sollozzo)

)

(:goal (and
      (dead lucabراس)
      (has-money-level sollozzo highmoney))
)
)

```

PROBLEM3.PDDL

```
(define (problem policiaco1) (:domain Policiaco)
(:objects

vito michael lucabراس alneri sollozzo captain-mccluskey captain-oneill
louis nino barzini - male
restaurant-louis jewelry-store-nino - business
kay amy marie - female
gun machine-gun knife shotgun revolver rifle - weapon
corleone-family barzini-family - organization
jealousy - jealousy
treason - treason
vengeance - vengeance
house-corleone house-alneri house-lucabراس house-sollozzo house-
barzini house-amy market train-station casino - genericplace
kidnapplace - kidnapplace
jail - jail
policestation - policestation
bank - bank
nomoney - nomoney
lowmoney - lowmoney
mediummoney - mediummoney
highmoney - highmoney

)

(:init

;Fijas-----

(next-money-level nomoney lowmoney)
(next-money-level lowmoney mediummoney)
(next-money-level mediummoney highmoney)

(is-police captain-mccluskey)
(is-police captain-oneill)

(has-business louis restaurant-louis)
(has-business nino jewelry-store-nino)

(boss-organization barzini barzini-family)
(boss-organization vito corleone-family)
(successor-organization michael corleone-family)
(is-boss vito)
(is-boss barzini)
(in-organization sollozzo barzini-family)
(in-organization michael corleone-family)
(in-organization alneri corleone-family)
(in-organization lucabراس corleone-family)

(lovers michael kay)
(lovers nino marie)
(father-son vito michael)

;Obligatorias y modificables-----
```

```

(at vito house-corleone)
(at michael house-corleone)
(at lucabراس house-lucabراس)
(at alneri house-alneri)
(at sollozzo market)
(at barzini house-barzini)
(at captain-mccluskey policestation)
(at captain-oneill train-station)
(at louis restaurant-louis)
(at marie jewelry-store-nino)
(at nino jewelry-store-nino)
(at amy house-amy)

(at gun restaurant-louis)
(at machine-gun house-corleone)
(at knife market)
(at shotgun house-sollozzo)
(at revolver house-lucabراس)
(at rifle house-barzini)

(has-money-level vito highmoney)
(has-money-level michael mediummoney)
(has-money-level lucabراس lowmoney)
(has-money-level alneri nomoney)
(has-money-level sollozzo nomoney)
(has-money-level barzini highmoney)
(has-money-level kay nomoney)
(has-money-level captain-mccluskey mediummoney)
(has-money-level captain-oneill nomoney)
(has-money-level louis highmoney)
(has-money-level marie lowmoney)
(has-money-level nino highmoney)
(has-money-level amy mediummoney)

;Añadibles-----

(motivation-money michael)
(motivation-money sollozzo)

(likes lucabراس amy)
(likes sollozzo amy)

)

(:goal (and
  (dead sollozzo)
  (at alneri jail)
  (kissed lucabراس amy))
)
)

```

PROBLEM4.PDDL

```
(define (problem policiaco1) (:domain Policiaco)
(:objects

vito michael lucabراس alneri sollozzo captain-mccluskey captain-oneill
louis nino barzini - male
restaurant-louis jewelry-store-nino - business
kay amy marie - female
gun machine-gun knife shotgun revolver rifle - weapon
corleone-family barzini-family - organization
jealousy - jealousy
treason - treason
vengeance - vengeance
house-corleone house-alneri house-lucabراس house-sollozzo house-
barzini house-amy market train-station casino - genericplace
kidnapplace - kidnapplace
jail - jail
policestation - policestation
bank - bank
nomoney - nomoney
lowmoney - lowmoney
mediummoney - mediummoney
highmoney - highmoney

)

(:init

;Fijas-----

(next-money-level nomoney lowmoney)
(next-money-level lowmoney mediummoney)
(next-money-level mediummoney highmoney)

(is-police captain-mccluskey)
(is-police captain-oneill)

(has-business louis restaurant-louis)
(has-business nino jewelry-store-nino)

(boss-organization barzini barzini-family)
(boss-organization vito corleone-family)
(successor-organization michael corleone-family)
(is-boss vito)
(is-boss barzini)
(in-organization sollozzo barzini-family)
(in-organization michael corleone-family)
(in-organization alneri corleone-family)
(in-organization lucabراس corleone-family)

(lovers michael kay)
(lovers nino marie)
(father-son vito michael)

;Obligatorias y modificables-----
```



```

(at vito house-corleone)
(at michael house-corleone)
(at lucabراس house-lucabراس)
(at alneri house-alneri)
(at sollozzo market)
(at barzini house-barzini)
(at captain-mccluskey policestation)
(at captain-oneill train-station)
(at louis restaurant-louis)
(at marie jewelry-store-nino)
(at nino jewelry-store-nino)
(at amy house-amy)
(at kay house-corleone)

(at gun restaurant-louis)
(at machine-gun house-corleone)
(at knife market)
(at shotgun house-sollozzo)
(at revolver house-lucabراس)
(at rifle house-barzini)

(has-money-level vito highmoney)
(has-money-level michael mediummoney)
(has-money-level lucabراس lowmoney)
(has-money-level alneri nomoney)
(has-money-level sollozzo nomoney)
(has-money-level barzini lowmoney)
(has-money-level kay lowmoney)
(has-money-level captain-mccluskey mediummoney)
(has-money-level captain-oneill nomoney)
(has-money-level louis highmoney)
(has-money-level marie lowmoney)
(has-money-level nino highmoney)
(has-money-level amy mediummoney)

;Añadibles-----

(debt kay barzini)
(debt louis vito)

(has-motive michael sollozzo vengeance)

(likes alneri marie)
(likes vito marie)

(motivation-money lucabراس)

)

(:goal (and
  (kidnapped-by kay barzini)
  (boss-organization michael corleone-family)
  (has-money-level lucabراس highmoney)
  (at lucabراس jail))
)
)

```

PROBLEM5.PDDL

```
(define (problem policiaco1) (:domain Policiaco)
(:objects

vito michael lucabراس alneri sollozzo captain-mccluskey captain-oneill
louis nino barzini - male
restaurant-louis jewelry-store-nino - business
kay amy marie - female
gun machine-gun knife shotgun revolver rifle - weapon
corleone-family barzini-family - organization
jealousy - jealousy
treason - treason
vengeance - vengeance
house-corleone house-alneri house-lucabراس house-sollozzo house-
barzini house-amy market train-station casino - genericplace
kidnapplace - kidnapplace
jail - jail
policestation - policestation
bank - bank
nomoney - nomoney
lowmoney - lowmoney
mediummoney - mediummoney
highmoney - highmoney

)

(:init

;Fijas-----

(next-money-level nomoney lowmoney)
(next-money-level lowmoney mediummoney)
(next-money-level mediummoney highmoney)

(is-police captain-mccluskey)
(is-police captain-oneill)

(has-business louis restaurant-louis)
(has-business nino jewelry-store-nino)

(boss-organization barzini barzini-family)
(boss-organization vito corleone-family)
(successor-organization michael corleone-family)
(is-boss vito)
(is-boss barzini)
(in-organization sollozzo barzini-family)
(in-organization michael corleone-family)
(in-organization alneri corleone-family)
(in-organization lucabراس corleone-family)

(lovers michael kay)
(lovers nino marie)
(father-son vito michael)

;Obligatorias y modificables-----
```

```

(at vito house-corleone)
(at michael house-corleone)
(at lucabراس house-lucabراس)
(at alneri house-alneri)
(at sollozzo market)
(at barzini house-barzini)
(at captain-mccluskey policestation)
(at captain-oneill train-station)
(at louis restaurant-louis)
(at marie jewelry-store-nino)
(at nino jewelry-store-nino)
(at amy house-amy)
(at kay house-corleone)

(at machine-gun house-corleone)
(at knife market)
(at shotgun house-sollozzo)
(at revolver house-lucabراس)
(at rifle house-barzini)

(has-money-level vito highmoney)
(has-money-level michael mediummoney)
(has-money-level lucabراس lowmoney)
(has-money-level alneri mediummoney)
(has-money-level sollozzo nomoney)
(has-money-level barzini mediummoney)
(has-money-level kay lowmoney)
(has-money-level captain-mccluskey mediummoney)
(has-money-level captain-oneill nomoney)
(has-money-level louis nomoney)
(has-money-level marie lowmoney)
(has-money-level nino highmoney)
(has-money-level amy mediummoney)

;Añadibles-----

(has-object sollozzo gun)

(has-motive sollozzo vito vengeance)

(motivation-money alneri)
(motivation-money louis)

(likes alneri kay)
(likes lucabراس amy)
(likes sollozzo amy)

)

(:goal (and
  (destroyed restaurant-louis)
  (dead captain-oneill)
  (dead michael)
  (at alneri jail)
)
)
)

```

PROBLEM6.PDDL

```
(define (problem policiaco1) (:domain Policiaco)
(:objects

vito michael lucabراس alneri sollozzo captain-mccluskey captain-oneill
louis nino barzini - male
restaurant-louis jewelry-store-nino - business
kay amy marie - female
gun machine-gun knife shotgun revolver rifle - weapon
corleone-family barzini-family - organization
jealousy - jealousy
treason - treason
vengeance - vengeance
house-corleone house-alneri house-lucabراس house-sollozzo house-
barzini house-amy market train-station casino - genericplace
kidnapplace - kidnapplace
jail - jail
policestation - policestation
bank - bank
nomoney - nomoney
lowmoney - lowmoney
mediummoney - mediummoney
highmoney - highmoney

)

(:init

;Fijas-----

(next-money-level nomoney lowmoney)
(next-money-level lowmoney mediummoney)
(next-money-level mediummoney highmoney)

(is-police captain-mccluskey)
(is-police captain-oneill)

(has-business louis restaurant-louis)
(has-business nino jewelry-store-nino)

(boss-organization barzini barzini-family)
(boss-organization vito corleone-family)
(successor-organization michael corleone-family)
(is-boss vito)
(is-boss barzini)
(in-organization sollozzo barzini-family)
(in-organization michael corleone-family)
(in-organization alneri corleone-family)
(in-organization lucabراس corleone-family)

(lovers michael kay)
(lovers nino marie)
(father-son vito michael)

;Obligatorias y modificables-----
```

```

(at vito house-corleone)
(at michael kidnapplace)
(at lucabراس house-lucabراس)
(at alneri house-alneri)
(at sollozzo market)
(at barzini house-barzini)
(at captain-mccluskey policestation)
(at captain-oneill train-station)
(at louis restaurant-louis)
(at marie jewelry-store-nino)
(at nino jewelry-store-nino)
(at amy house-amy)
(at kay house-corleone)

(at machine-gun house-corleone)
(at knife market)
(at shotgun house-sollozzo)
(at revolver house-lucabراس)
(at rifle house-barzini)

(has-money-level vito highmoney)
(has-money-level michael mediummoney)
(has-money-level lucabراس lowmoney)
(has-money-level alneri mediummoney)
(has-money-level sollozzo nomoney)
(has-money-level barzini lowmoney)
(has-money-level kay highmoney)
(has-money-level captain-mccluskey mediummoney)
(has-money-level captain-oneill nomoney)
(has-money-level louis mediummoney)
(has-money-level marie lowmoney)
(has-money-level nino highmoney)
(has-money-level amy mediummoney)

;Añadibles-----

(kidnapped-by michael barzini)
(non-movable michael)

(bodyguard lucabراس vito)
(has-bodyguard vito)

(has-object alneri gun)

(has-motive barzini vito vengeance)

(likes sollozzo marie)

(debt louis lucabراس)
(debt michael barzini)
)
(:goal (and
  (boss-organization michael corleone-family)
  (at nino jail)
  (kissed sollozzo marie)
  (dead lucabراس)
  (at michael policestation))))

```

PROBLEM7.PDDL

```
(define (problem policiaco1) (:domain Policiaco)
(:objects

vito michael lucabradi alneri sollozzo captain-mccluskey captain-oneill
louis nino barzini vincenzo johnny anthony falcone sonny - male
restaurant-louis jewelry-store-nino - business
kay amy marie - female
gun machine-gun knife shotgun revolver rifle - weapon
corleone-family barzini-family - organization
jealousy - jealousy
treason - treason
vengeance - vengeance
house-corleone house-barzini train-station casino - genericplace
kidnapplace - kidnapplace
jail - jail
policestation - policestation
bank - bank
nomoney - nomoney
lowmoney - lowmoney
mediummoney - mediummoney
highmoney - highmoney
)

(:init

;Fijas-----

(next-money-level nomoney lowmoney)
(next-money-level lowmoney mediummoney)
(next-money-level mediummoney highmoney)

(is-police captain-mccluskey)
(is-police captain-oneill)

(has-business louis restaurant-louis)
(has-business nino jewelry-store-nino)

(boss-organization barzini barzini-family)
(boss-organization vito corleone-family)
(successor-organization michael corleone-family)
(successor-organization sollozzo barzini-family)
(is-boss vito)
(is-boss barzini)
(in-organization sollozzo barzini-family)
(in-organization johnny barzini-family)
(in-organization michael corleone-family)
(in-organization alneri corleone-family)
(in-organization lucabradi corleone-family)
(in-organization sonny corleone-family)

(lovers michael kay)
(lovers nino marie)
(father-son vito michael)
(father-son vito sonny)
```

;Obligatorias y modificables-----

(at vito house-corleone)
(at michael house-corleone)
(at lucabراسي train-station)
(at alneri casino)
(at sollozzo house-barzini)
(at barzini house-barzini)
(at captain-mccluskey policestation)
(at captain-oneill train-station)
(at louis restaurant-louis)
(at marie jewelry-store-nino)
(at nino jewelry-store-nino)
(at amy casino)
(at kay house-corleone)
(at vincenzo train-station)
(at johnny house-barzini)
(at anthony restaurant-louis)
(at falcone casino)
(at sonny house-corleone)

(at machine-gun house-corleone)
(at knife casino)
(at shotgun house-barzini)
(at revolver train-station)
(at rifle jewelry-store-nino)

(has-money-level vito mediummoney)
(has-money-level michael nomoney)
(has-money-level lucabراسي nomoney)
(has-money-level alneri nomoney)
(has-money-level sollozzo mediummoney)
(has-money-level barzini highmoney)
(has-money-level kay highmoney)
(has-money-level captain-mccluskey mediummoney)
(has-money-level captain-oneill nomoney)
(has-money-level louis nomoney)
(has-money-level marie lowmoney)
(has-money-level nino highmoney)
(has-money-level amy mediummoney)
(has-money-level vincenzo nomoney)
(has-money-level johnny mediummoney)
(has-money-level anthony lowmoney)
(has-money-level falcone highmoney)
(has-money-level sonny mediummoney)

;Añadibles-----

(debt vincenzo vito)

(likes falcone kay)
(likes sollozzo amy)
(likes anthony amy)
(likes sonny amy)
(likes alneri marie)

(commanded-to-kill sonny anthony vito)

```
(motivation-money vincenzo)
(motivation-money johnny)
(motivation-money louis)

(bodyguard johnny barzini)
(has-bodyguard barzini)
(bodyguard lucabراسي vito)
(has-bodyguard vito)

)
(:goal (and
  (dead louis)
  (dead vito)
  (dead anthony)
  (dead captain-oneill)
  (dead kay)
  (at michael jail)
  (has-money-level vincenzo highmoney)
  (at nino jail)
  (destroyed restaurant-louis)
)
)
)
```